

Encoding and decoding a message within an image

Patent number: DE19983484T
Publication date: 2001-07-26
Inventor: YIP PHILIP S (US)
Applicant: CHANG KENNETH H P (US)
Classification:
- international: **G06K7/14; G06K19/06; G06K7/14; G06K19/06; (IPC1-7): G06K19/06**
- european: **G06K7/14; G06K19/06C5**
Application number: DE19991083484T 19990717
Priority number(s): US19980138591 19980822; WO1999US12376 19990717

Also published as:



WO0011599 (A1)
WO0011599 (A1)
US6256398 (B1)

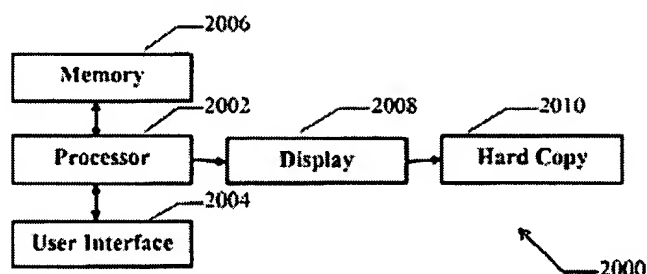
Best Available Copy

Report a data error here

Abstract not available for DE19983484T

Abstract of corresponding document: **US6256398**

A method for decoding a message embedded in a pattern of pixels. The method includes the steps of determining the pixel values for pixels from the pattern of pixels, determining binary values from the pixel values for pixels from the pattern of pixels; and determining the embedded message from the binary values. The pixels have a range of pixel values between a maximum and a minimum. The pixels are divided into cells each having glyph cell and background pixels. The binary value of a glyph pixel is determined by the contrast the glyph pixel has with its background pixels. The method can be used to decode embedded web-site address from an image with a foreground image and the embedded web-site address.



Data supplied from the **esp@cenet** database - Worldwide



①9 BUNDESREPUBLIK
DEUTSCHLAND



DEUTSCHES
PATENT- UND
MARKENAMT

⑫ **Veröffentlichung**
⑩ **DE 199 83 484 T 1**

⑤① Int. Cl.⁷:
G 06 K 19/06

- der internationalen Anmeldung mit der
- ⑧⑦ Veröffentlichungsnummer: WO 00/11599 in
deutscher Übersetzung (Art. III § 8 Abs. 2 IntPatÜG)
- ②① Deutsches Aktenzeichen: 199 83 484.9
- ⑧⑥ PCT-Aktenzeichen: PCT/US99/12376
- ⑧⑥ PCT-Anmeldetag: 17. 7. 1999
- ⑧⑦ PCT-Veröffentlichungstag: 2. 3. 2000
- ④③ Veröffentlichungstag der PCT-Anmeldung
in deutscher Übersetzung: 26. 7. 2001

DE 199 83 484 T 1

- ③⑩ Unionspriorität:
09/138,591 22. 08. 1998 US
- ⑦① Anmelder:
Chang, Kenneth H.P., Foster City, Calif., US
- ⑦④ Vertreter:
Müller, Schupfner & Gauger, 80539 München

- ⑦② Erfinder:
Yip, Philip S., Foster City, Calif., US

- ⑤④ Codierung und Decodierung einer Nachricht in einem Bild

199 83 484 T 1

20.09.01

DE 199 83 4 84 T 1

Kenneth H.P. Chang

8983.PT-DE

PM/TE

Codierung und Decodierung einer Nachricht in einem Bild

Gebiet der Erfindung

Die vorliegende Erfindung bezieht sich auf Verfahren zum Codieren und Decodieren von Nachrichten und insbesondere auf Verfahren zum Codieren einer Nachricht in ein Bild, wobei die Nachricht nicht durch visuelle Inspektion decodiert werden kann.

Hintergrund der Erfindung

Ein maschinenlesbarer Code auf der Produktverpackung ist eine gute Möglichkeit, um produktbezogene Informationen zu übermitteln. So wurde beispielsweise viele Jahre der Barcode verwendet, um Information, wie z.B. Produktidentifizierungs- und Lagerinformation zu beinhalten. Solche Barcodes können optisch gelesen werden, um die in den Barcodes codierte Information wiederherzustellen.

Barcodes sind jedoch eindimensional und sind in der Informationsmenge, die sie speichern können, begrenzt. Demzufolge wurde eine zweidimensionale Symbologie entwickelt, um die Datenmenge, die in solchen Codes gespeichert werden kann, zu erhöhen. Bei der Verwendung von zweidimensionalen Codes zum Registrieren von Information ist eine präzise Synchronisation nötig, um aufeinanderfolgend die Daten von dem Symbolmuster les-

20001

DE 199 83 484 T1

sen zu können. Um eine Orientierung für die zweidimensionale Symbologie zu liefern, benötigen die Codierverfahren oft visuell identifizierbare Merkmale wie Linien, Rahmen, konzentrische Ringe, Achsen, Spalten oder Symbolreihen o.ä., die optisch von anderen Symbolen und Bildern unterschieden werden können. Leider sind solche Verfahren alles andere als wünschenswert, wenn die Information in ein visuelles Bild eingebettet werden soll, da die visuell identifizierbaren Merkmale für einen Betrachter störend sind, der das Bild ohne Ablenkung betrachten möchte.

Benötigt wird ein Verfahren zum Codieren und Decodieren von eingebetteten Nachrichten in einem visuellen Bild ohne störende Merkmale, die die eingebettete Nachricht darstellen oder die Synchronisation oder die Orientierung des Datenmusters.

Die US Patente U.S.P.N. 4,263,504 (Thomas), U.S.P.N. 5,189,292 (Batterman et al), U.S.P.N. 5,128,525 (Stearns et al.), U.S.P.N. 5,221,833 (Hecht), U.S.P.N. 5,245,165 (Zhang), U.S.P.N. 5,329,107 (Priddy et al.), U.S.P.N. 5,439,354 (Priddy), U.S.P.N. 5,481,103 (Wang), U.S.P.N. 5,507,527 (Tomiooka, et al.), U.S.P.N. 5,515,447 (Zheng, et al), U.S.P.N. 5,521,372 (Hecht et al.), U.S.P.N. 5,541,396 (Rentsch), U.S.P.N. 5,572,010 (Petrie), U.S.P.N. 5,576,532 (Hecht) und U.S.P.N. 5,591,956 (Longacre, et al.) sind Beispiele für Beschreibungen der Verwendung einer zweidimensionalen Symbologie beim Speichern codierter Information. Die Beschreibungen dieser zitierten Patente werden hiermit durch Verweis aufgenommen.

200201
-3-

DE 199 83 484T1

Zusammenfassung

Die vorliegende Erfindung stellt Verfahren zum Decodieren einer Nachricht, die in einem Pixelmuster eingebettet ist, bereit. Das Verfahren beinhaltet Bestimmen der Pixelwerte für Pixel von dem Pixelmuster; Bestimmen der binären Werte aus den Pixelwerten für Pixel aus dem Pixelmuster; und Bestimmen der eingebetteten Nachricht aus den binären Werten. In einem anderem Aspekt stellt die Erfindung eine Vorrichtung bereit zum Codieren des Bildes, wobei ein Vordergrundbild eine eingebettete Nachricht hat, und stellt ebenfalls eine Vorrichtung zum Decodieren des Bildes für die eingebettete Nachricht bereit.

Die Erfindung der vorliegenden Erfindung kann vorteilhafterweise für das Codieren und Decodieren von Nachrichten in einem Vordergrundbild verwendet werden, ohne daß es zu störenden Veränderungen im Bild kommt, die den Betrachter zu sehr davon ablenken könnten, das Vordergrundbild visuell wahrzunehmen. Ein Beispiel ist das Laden einer URL-Adresse (die in das visuelle Bild eingebettet wurde) in einen Internetbrowser. Wenn eine Person im "Netz surft" und auf einen "Hotspot" kommt (der oft eine photographische Darstellung oder Wörter in blauer Schrift darstellt), der mit einer anderen Website durch Hyperlink verbunden ist, benutzt die Person einfach die Zeigevorrichtung (Maus) und klickt auf den Hotspot, um auf die Website dieses Hotspots zu gelangen. Es gibt jedoch keine einfache Möglichkeit, das dargestellte Bild auszudrucken und das Bild später zu benutzen, um die URL-Adresse dieser Website direkt in einen Browser einzugeben, um mit der Website verbunden zu werden. Die vorliegende Erfindung stellt ein Verfahren bereit sowie eine Vorrichtung, um es zu ermöglichen, eine Website zu drucken, um das visuelle Bild zu erhalten und es dennoch einem elektronischen Lesegerät zu ermöglichen, das Bild zu lesen und den Internetbrowser so zu steuern, daß er mit dieser Website

verbindet. Der Prozeß beinhaltet das Erhalten eines Lichts von einem Display, der ein Pixelmuster anzeigt, und Bestimmen des Pixelwerts eines jeden Pixels, Wiederherstellen einer eingebetteten URL-Adresse durch Bestimmen binärer Werte aus den Pixelwerten der Pixel; und Laden der URL-Adresse in den Internetbrowser zum Verbinden mit der Website gemäß der URL-Adresse. Die vorliegende Erfindung kann Bilder in Graustufen oder vielfarbige Bilder verwenden, um eingebettete Nachrichten in Bildern zu kommunizieren. So können auf einer visuellen Ebene Bilder und Wörter so dargestellt werden, daß sie ein Mensch erkennen und lesen kann, aber auf einer weniger offensichtlichen Ebene kann eine im Bild eingebettete Nachricht von einer Maschine gelesen werden. Das Bild mit der eingebetteten Nachricht kann, wenn es in ausgedruckter Form dargestellt ist, auf bequeme Art und Weise transportiert werden.

Kurze Beschreibung der Zeichnungen

Die folgenden Figuren sind zur besseren Veranschaulichung der Ausführungsformen der Vorrichtung und Verfahren der vorliegenden Erfindung beigefügt. In diesen Figuren stellen gleiche Nummerierungen gleiche Merkmale in verschiedenen Ansichten dar.

Fig. 1 zeigt eine Ausführungsform einer graustufigen Darstellung eines Pixelzellblocks.

Fig. 2 zeigt eine Ausführungsform einer Kachel, die vier Pixelzellen hat.

Fig. 3A zeigt eine Zelle mit 3 x 3 Pixeln.

Fig. 3B zeigt eine Zelle mit 5 x 5 Pixeln.

Figs. 4A bis 4I zeigen Ausführungsformen, wie Pixel in Zellen die logische "1" codieren.

Figs. 5A bis 5I zeigen Ausführungsformen, wie Pixel in Zellen die logische "0" codieren.

20.00.01

DE 199 83 484 T1

Figs. 6a bis 6I zeigen Ausführungsformen wie die Pixel in farbigen Zellen die logische "1" und die logische "0" codieren.

Fig. 7 zeigt ein Bild aus Pixeln, das das Wort "Webstar" darstellt.

Fig. 8A zeigt ein Glyphbild, das das Wort "Webstar" mit einer eingebetteten Nachricht darstellt.

Fig. 8B zeigt eine Glyphabbildung oder Glyphkarte (glyph map), die dem Glyphbild aus Fig. 8A entspricht.

Fig. 8C zeigt eine Zellenabbildung oder Zellkarte (cell map), die dem Glyphbild aus Fig. 8A entspricht.

Fig. 8D zeigt eine Rasterabbildung oder Rasterkarte (grid map), die dem Glyphbild aus Fig. 8A entspricht.

Fig. 8E zeigt eine Datenkarte (data map), die Synchronisationszellen und Datenzellen hat.

Fig. 9A stellt in Form eines Ablaufdiagramm kurz den Prozeß der Codierung einer eingebetteten Nachricht dar.

Fig. 9B stellt eine Ausführungsform des Codierverfahrens in Blockform dar.

Fig. 10 zeigt Selbstsymmetrie, einschließlich Glyphzellen, die symmetrisch um die globale Zentrumszelle liegen.

Fig. 11 zeigt eine Ausführungsform mit einem einschichtigen Datenblockzellenlayout und symmetrischen Datenzellen.

Fig. 12A zeigt eine Ausführungsform eines doppelschichtigen Datenblockzellenlayout mit Synchronisationszellen und symmetrischen Datenzellen.

Fig. 12B zeigt einen Vorwärtssynchronisationsstrom mit Blockfahne und Fehlerkorrekturcode und

Fig. 12C zeigt einen Rückwärtssynchronisationsstrom, der ein Spiegelbild des Vorwärtssynchronisationsstroms aus Fig. 12B ist.

Fig. 12D zeigt einen Vorwärtsdatenstrom mit Nachrichtendaten und Fehlerkorrekturcode und

20.02.01

DE 199 83 484 T1

Fig. 12E zeigt einen Rückwärtsdatenstrom, der ein Spiegelbild des Vorwärtsdatenstroms aus Fig. 12D ist.

Fig. 13 zeigt eine Ausführungsform eines doppelschichtigen Datenblockzellayouts mit symmetrischen Synchronisationszellen (Synczellen) und asymmetrischen Datenzellen.

Fig. 14A zeigt schematisch eine Ausführungsform eines Bildes, das aus seriellen Blöcken abgeleitet wurde.

Figs. 14B bis 14D zeigen die seriellen Blöcke für das Bild aus Fig. 14A.

Fig. 14E zeigt den resultierenden Block, der aus den seriellen Blöcken 14B bis 14D resultiert.

Fig. 14F zeigt eine schematische Darstellung einer Ausführungsform von vier parallelen Blöcken.

Figs. 14G bis Fig. 14J zeigen die individuellen Blöcke der vier parallelen Blöcke aus Fig. 14F.

Fig. 14K ist das gesamte Bild des resultierenden Blocks, der aus den parallelen Blöcken von Fig. 14F besteht.

Fig. 15 zeigt ein Beispiel der Codes einer HTML-Seite

Fig. 16 zeigt das Erscheinungsbild von typischen konventionellen "Hotspots" für die Websites der HTML-Seite aus Fig. 15

Fig. 17 zeigt das Erscheinungsbild eines Bildes mit eingebetteten URL-Sites-"Hotspots" für die Websites dieser HTML-Seite aus Fig. 15.

Fig. 18 zeigt eine in sich symmetrische, doppelschichtige, syncsymmetrische und datensymmetrische Datenkarte.

Fig. 19A stellt als Ablaufdiagramm kurz den Prozeß des Decodierens und Antwortens auf eine eingebettete Nachricht dar.

Fig. 19B stellt eine Ausführungsform des Decodierverfahrens in Blockform dar.

Fig. 20A zeigt eine Ausführungsform einer Vorrichtung zum Codieren einer Nachricht.

20.02.01

DE 199 83 484 T1

Fig. 20B stellt eine Ausführungsform einer Lesevorrichtung zum Lesen der eingebetteten Nachricht dar.

Fig. 20C zeigt eine Ausführungsform der vorliegenden Erfindung zum Decodieren einer gedruckten Seite mit einer eingebetteten URL-Adresse.

Fig. 20D zeigt schematisch die Struktur einer Lesevorrichtung.

Fig. 21A zeigt ein Ablaufdiagramm, das darstellt wie eine URL-Adresse in ein Glyphbild eingebettet wird.

Fig. 21B zeigt ein Ablaufdiagramm, das darstellt, wie eine URL-Adresse aus einem Glyphbild decodiert wird.

Detaillierte Beschreibung

In einem Aspekt der Erfindung stellt die vorliegende Erfindung ein Verfahren zum Einbetten einer Nachricht in ein visuelles Bild ohne störende Merkmale, die die Aufmerksamkeit vom visuellen Bild ablenken, bereit. Ein Beispiel dafür ist das Einbetten einer Hyperlink-Adresse wie einer URL-Adresse ("Website-Adresse") in ein Bild, das auf Papier gedruckt ist.

Als erste Darstellung wird eine Ausführungsform eines einfarbigen Bildes (oder graustufigen Bildes) mit einer eingebetteten Nachricht im folgenden näher beschrieben. Der Fachmann wird allerdings in der Lage sein, die Verfahren der vorliegenden Offenbarung einfach für eine farbige Ausführungsformen zu adaptieren. Für eine Farbe (z.B. schwarz-grau-weiß) in einem graustufigen Bild haben die Pixel Grauwerte, die von einem minimalen Pixelwert (der z.B. schwarz darstellt) bis zu einem maximalen Pixelwert (der z.B. weiß darstellt) reichen. Ebenso können für farbige Ausführungsformen in anderen Farben, die nicht schwarz-weiß sind, Farbabstufungsbereiche auf ähnliche Art und Weise implementiert werden.

20.02.01

DE 199 83 484T1

Einfarbiges Bild

Um die vorliegende Erfindung und die schwarz-weiße, graustufige Ausführungsform besser illustrieren zu können, werden die folgenden Werte angenommen. Es ist offensichtlich, daß andere Werte (z.B. Farbe, Größe, Dimensionen u.ä.) einfach von einem Fachmann adaptiert werden können.

Figur 1 zeigt eine Ausführungsform einer graustufigen Darstellung eines Blocks 100 aus Zellen 112 von Pixeln. Die Zellen werden in zweidimensionale Gruppen, die "Kacheln" 122 genannt werden, unterteilt. Die codierte Information in einem Bild wird von Symbolen oder "Glyphs" repräsentiert, die in einer schwarz-weißen Ausführungsform durch Pixel 110 repräsentiert werden. Um die logische Information zu transportieren, sind die Pixel 110 in Glyphzellen (im nachfolgenden einfach "Zellen") 112 unterteilt, von denen jede ein quadratisches ($n \times n$) Feld ist, um einen binären Bitwert zu transportieren. Der Fachmann weiß, daß mehr als eine Zelle verwendet werden kann, um einen binären Bit zu transportieren, indem jede Zelle eine Komponente oder einen Bruchteil eines Bits darstellt. Beispiele sind das Bestimmen des Bits als ganzes aus der Summe, dem Querschnitt, dem Maximum, dem Kontrast, der Ähnlichkeit o.ä. aus zwei oder mehr Zellen. Aus Gründen der Verständlichkeit der Beschreibung wird jedoch in dieser graustufigen Ausführungsform, wenn es nicht speziell vermerkt wird, eine Zelle ein Bit bedeuten. Eine Zelle kann eine Synchronisationszelle (Synczelle) 124 oder eine Datenzelle 126 (s. Figur 2) sein. (Die Punkte in Figuren 1 und 2 in der oberen linken Zelle in einer Kachel sind in die Figuren aufgenommen worden, um es dem Leser dieser Offenbarung zu erleichtern, die Synczellen zu identifizieren, und bezeichnen nicht notwendigerweise beobachtbare Unterschiede). Wie aus Figuren 2 und 3A ersichtlich

20.02.01

DE 199 83 484 T1

ist, repräsentiert unter den Pixeln in einer Zelle 112 (oder Synczelle 124 oder Datenzelle 126), ein Pixel, das "Glyph Pixel" (GP Pixel) 116 (s. Figur 3A) den logischen Wert (d.h. den binären Wert) der Zelle, während die anderen Pixel (die Hintergrundpixel oder "Background Pixel" "BP Pixel") 118 in der Nähe (in dieser Ausführungsform befinden sie sich um das GP herum) als Hintergrund dienen und die Bestimmung des primären Pixels beim Codieren als "schwarz", "grau" oder "weiß", abhängig von den Pixelwerten aller Pixel in der Zelle, erleichtern. Wie diese Bestimmung ausgeführt wird, wird im nachfolgenden beschrieben. Es gibt verschiedene Graustufenebenen. Zur Illustration wird die 3 x 3 Zelle verwendet, um die vorliegende Erfindung zu beschreiben, der Fachmann kann jedoch ganz einfach andere Zellgrößen (z.B. $n = 2, 4, 5$ usw.) basierend auf der vorliegenden Erfindung implementieren. Es ist auch nicht nötig, daß die Zelle ein Quadrat ist. In einer 3 x 3 Zelle, wie in Figuren 2 und 3A dargestellt, sind die Hintergrundpixel BP1, BP2, BP3, BP4, BP6, BP7, BP8 und BP9.

Die logische Information in dem Bild wird durch die logischen binären Bitwerte 0 und 1 repräsentiert. In den Pixel, die die logische Information transportieren, repräsentiert der Kontrast zwischen einem Pixel und den anderen Pixeln (d.h. zwischen dem GP-Pixel 116 und den BP-Pixeln 118) z.B. in einer Zelle, ein Bit.

Der Pixelwert von einem Pixel (egal ob es ein GP-Pixel oder ein BP-Pixel ist) ist ein Grauwert zwischen einem Minimum und einem Maximum, das üblicherweise zwischen 0 und 255 liegt (0 = schwarz und 255 = weiß).

Wie in einer Ausführungsform in Figuren 1 und 2 dargestellt, sind die Zellen 112 in "Glyphkacheln" (im nachfolgenden ein-

20.00.01
DE 199 83 48411

fach als "Kachel" bezeichnet), von denen jede $m \times r$ Zellen beinhaltet, und wobei m und r ganz Zahlen sind, angeordnet. Zur Illustration enthält jede Kachel in der Ausführungsform 2×2 Zellen. Es ist jedoch nicht nötig, daß die Kachel ein Quadrat ist. Figur 2 zeigt eine Kachel, die der in Block 100 in Figur 1 ähnlich ist. Eine der Zellen in der Kachel 122 ist eine Synczelle 124 und der Rest sind Datenzellen 126. Der logische Wert jeder Zelle 122 wird durch den Kontrast des GP-Pixels zu den BP-Pixeln repräsentiert und ist entweder ein Datenbit, das die eingebettete Nachricht und die assoziierte Fehlerkorrektur der Nachricht repräsentiert, oder ein Synchronisationsbit (Syncbit), das die Interpretation der Daten steuert, z.B. indem es die Länge der Nachricht bestimmt, oder die Stellen der Fehlerkorrekturbits u.ä. So beinhaltet jede Kachel 122 eine Synczelle 124 und $(m \times r - 1)$ Datenzellen 126. Die Synczellen 124 stellen eine räumliche Synchronisation bereit, um die logische Ordnung der Daten (die in den Datenzellen 126 enthalten sind) zu erhalten, so daß der Glyphcode gelesen und decodiert werden kann. Zu diesem Zweck sind die Synczellen 124 und die Datenzellen 126 auf vorbestimmte Art und Weise angeordnet, z.B. in regelmäßigem Abständen, die sich von Kachel zu Kachel wiederholen. Figuren 1 und 2 zeigen die Position der Synczelle 124 relativ zu den Datenzellen 126 in einer Ausführungsform. In dieser Ausführungsform ist die Synczelle 124 aus der Sicht des Betrachters in der oberen linken Ecke einer Kachel, die auch Datenzellen 126A, 126B und 126C enthält, angeordnet. In dieser Ausführungsform sind die Datenbits beispielsweise so angeordnet, daß das erste Bit in der Datenzelle 126A angeordnet ist, das zweite Bit in der Datenzelle 126B und das dritte Bit in der Datenzelle 126C in der ersten Kachel. Die nächsten drei Datenbits sind in den Datenzellen 126A, 126B und 126C der zweiten Kachel angeordnet und so weiter. Kleinere Synchronisationsfehler können durch das Bereitstellen einer Fehlerkorrek-

20.02.01
DE 199 83 484 T1

turmöglichkeit toleriert werden. Diese Ausführungen sind so zu verstehen, daß die Anordnung der Synczelle in Bezug auf die Datenzellen nur zu Illustration dient. Andere Arten der Anordnung sind möglich. So benötigt z.B. die in sich symmetrische, einschichtige Anordnung, wie sie im nachfolgenden beschrieben wird, keine Synczellen außer den Datenzellen.

Das Codieren von Glyphzellen

Im allgemeinen können m GPs und n BPs in einer Zelle sein, wobei n größer als m ist (s. Figur 3B). Jede Zelle hat einen Zellenkontrast (cell contrast, CC), der eine Funktion der Pixelwerte von allen Pixeln der Zelle ist.

$$CC = ABS ((GP1 + GP2 + \dots + GPM)/m - (BP1 + BP2 + \dots + BPN)/n)$$

wobei ABS in dieser Gleichung die absolute Funktion ist, GP1, GP2 und GPM die Pixelwerte der Glyphpixel darstellen und BP1, BP2, BPM die Pixelwerte der Hintergrundpixel bedeuten.

Aus Gründen der Einfachheit hat in Figur 3A bei den illustrativen 3 x 3 großen monotonen (d.h. schwarz-weiß-graustufigen) Glyphzellen, eine Glyphzelle 126 nur ein zentrales Glyphpixel, GP, und acht Hintergrundpixel (BP1, BP2, BP3, BP4, BP6, BP7, BP8, BP9), die um das Glyphpixel herum angeordnet sind.

$$CC = ABS (GP - (BP1 + BP2 + BP3 + BP4 + BP6 + BP7 + BP8 + BP9)/8)$$

wobei ABS in dieser Gleichung die absolute Funktion ist und GP, BP1, BP2 etc. die Pixelwerte der entsprechenden Pixel darstellen. Schwarz hat den Pixelwert 0. Weiß hat den Pixelwert 255. Die Pixelwerte und damit der Zellenkontrast können von 0 bis 255 variieren. Um eine eingebettete Nachricht zu repräsen-

tieren kann der Zellenkontrast durch Ändern des Wertes des GP Pixel geändert werden. Die Werte der Hintergrundpixel BP1, BP2 ... BP9 ändern sich vorzugsweise nicht, wodurch die Verzerrung des originalen visuellen Bildes (Vordergrundbildes), das die Pixelwerte der Hintergrundpixel bestimmt, gering bleibt.

Um ein binäres Bit mit "1" zu codieren, kann man den Wert des Glyphpixels ändern, so daß der assoziierte Zellkontrast groß ist, vorzugsweise so groß wie möglich (wobei vorzugsweise die Hintergrundpixel wie bereits erwähnt konstant gehalten werden). Um andererseits ein digitales Bit mit "0" zu codieren, kann man den Pixelwert des GP-Pixel so ändern, daß der Zellenkontrast sehr klein ist, vorzugsweise so klein wie möglich. Auf diese Art und Weise können die zwei binären Werte leicht unterschieden werden. Aufgrund der Verwendung der absoluten Funktion, kann der binäre Wert weiterhin auf dieselbe Art und Weise codiert werden, unabhängig davon, ob der Hintergrund hell oder dunkel ist. Selbstverständlich ist die Wahl des Codierens einer "1" und einer "0" auf die oben beschriebene Weise beliebig. Man kann die beiden binären Werte auch anders herum codieren.

Eine Möglichkeit der Ausführung des oben beschriebenen Verfahrens der Maximierung des Zellkontrasts für "1" und der Minimierung des Zellkontrasts für "0" ist wie folgt. Ein Zellgewicht (Cell weight, CW) wird aus den Pixelwerten der Hintergrundpixel berechnet.

Um logische Werte zu codieren, werden die folgenden Schritte verwendet:

A. Definiere die Codierschwelle (Encoding Threshold, ET) (es ist zu beachten, daß ET unabhängig von Glyphzelle, Glyphkachel und Glyphblock ist)

20.02.01

DE 199 83 484 T 1

$$ET = (\text{Schwarz} + \text{Weiß})/2$$

ET ist der durchschnittliche Pixelwert von Schwarz (dessen Pixelwert 0 ist) und Weiß (dessen Pixelwert 255 ist). ET hat also einen Graustufenwert von 127.

B. Berechne das durchschnittliche Zellgewicht (Cellweight, CW), zur Verwendung zur Hintergrundumkehr, wie im nachfolgenden für die Codierung des GP-Pixelwerts beschrieben.

$$CW = (BP1 + BP2 + BP3 + BP4 + BP6 + BP7 + BP8 + BP9)/8$$

wobei BP1 bis BP9 in dieser Gleichung den Pixelwert, den sie darstellen, repräsentieren (d.h. jeweils des Pixels BP1 bis BP9).

C. Berechne den Glyphpixelwert

Um den GP mit der logischen eins (1) zu codieren, wird dem GP ein Pixelwert gegeben, der die "Umkehrung" des durchschnittlichen Zellgewichts ist. Wenn, in anderen Worten, das allgemeine Zellgewicht mehr weiß als schwarz ist, dann wird das GP-Pixel auf mehr schwarz als weiß gesetzt, vorzugsweise auf schwarz:

$$GP = \text{Schwarz wenn } CW > ET \text{ oder } CW = ET$$

$$GP = \text{Weiß wenn } CW < ET$$

Wenn das Glyphpixel also den Pixelwert umkehrt, um die logische 1 zu codieren, bekommt es vorzugsweise den extremen Pixelwert (d.h. der Pixelwert ist entweder am hellsten 255 oder am dunkelsten 0). Auf diese Art und Weise ist der Kontrast zwischen dem Pixelwert des Glyphpixels und seiner Umgebung nach dem Codieren der logischen 1 am stärksten. Der Zellkon-

20.02.01
DE 199 83 484 T1

trast CC ist also nach dem Codieren einer logischen "1" am größten.

Figur 4A zeigt eine 3x3 Zelle bevor ein GP mit der logischen 1 codiert wurde, wobei der Hintergrund weiß ist. So haben die Hintergrundpixel BP1 bis BP9 alle den Pixelwert 255, genauso wie das GP-Pixel. Figur 4I ist die Legende (ein Schlüssel zu den Graustufen) für Figuren 4A bis 4H. Figur 4B zeigt die Zelle nach dem Codieren einer 1. Nach dem Codieren haben die Hintergrundpixel BP1 bis BP9 immer noch den Pixelwert von 255, während sich das Glyphpixel von weiß nach schwarz ändert und einen Pixelwert von 0 hat, was die größtmögliche Differenz vom durchschnittlichen Hintergrund ist. Figur 4C zeigt eine 3 x 3 Zelle vor dem Codieren des GP-Pixel mit einer logischen 1, wenn der Hintergrund schwarz ist (BP1 bis BP9 haben alle einen Pixelwert von 0). Figur 4D zeigt die Zelle aus Figur 4C, nachdem die 1 codiert wurde. Das Glyphpixel hat nun einen Pixelwert von 255. Figuren 4E und 4F zeigen entsprechend eine 3 x 3 Zelle vor und nach dem Codieren des GP mit der logischen 1, wenn der Hintergrund graue Pixel (BP1, BP4 sind hellgrau, BP7 ist grau, BP8 und BP9 sind dunkelgrau) und schwarze Pixel (BP2, BP3, BP6 haben jeweils einen Pixelwert von 0) enthält. Der Hintergrund ist im Durchschnitt also mehr schwarz als weiß. Folglich ändert sich das GP von schwarz nach weiß, um eine logische 1 zu codieren. Figuren 4G und 4H zeigen eine 3 x 3 Zelle vor und nach dem Codieren eines GP mit der logischen 1, wobei der Hintergrund weiße Pixel (BP2, BP3, BP6), hellgrau Pixel (BP1, BP4) und graue Pixel (BP7, BP8, BP9) enthält, wodurch der durchschnittliche Hintergrund mehr weiß als schwarz ist. Um die logische 1 zu codieren, ändert sich das Glyphpixel vom ursprünglichen Wert von weiß zu schwarz.

20.02.01
DE 199 83 484 T1

Um die logische 1 zu codieren, bewegt sich das Glyphpixel zu einem postcodierten Pixelwert, der soweit wie möglich vom durchschnittlichen Hintergrund entfernt ist. Um jedoch ein Datenbit mit null (0) zu codieren, bleibt der GP-Pixelwert nahe dem des durchschnittlichen Hintergrunds. In einer Ausführungsform bleibt der GP-Pixelwert unverändert vom Wert vor der Codierung. Dieses Verfahren ist gut geeignet, wenn die Graustufe des GP nicht unterschiedlich von dem der umgebenden Pixel ist, die das Vordergrundbild zeigen, was normalerweise der Fall ist bei makroskopischen visuellen Vordergrundbildern, die durch kleine Pixel gezeigt werden. In diesen Fällen bleibt der Pixelwert des Glyphpixels in der Mitte der Zelle unverändert, um die logische 0 zu codieren. Der Unterschied zwischen dem GP nach dem Codieren der logischen 0 und dem durchschnittlichen Hintergrund ist typischerweise sehr klein. Deshalb repräsentiert ein unverändertes Pixel entweder eine logische 0 oder ein Hintergrundpixel.

Eine Alternative ist, den postcodierten Pixelwert des GP aktiv so zu setzen, daß er nahe, vorzugsweise gleich dem durchschnittlichen Hintergrund ist. In dieser bevorzugten Ausführungsform ist der Unterschied der Pixelwerte zwischen dem durchschnittlichen Hintergrund und dem GP null nach dem Codieren der logischen 0. Wenn man diese Nulldifferenz mit der Differenz von wenigstens 127 Pixelwerten zum Codieren einer logischen 1 vergleicht, kann man einfach sagen, ob ein GP eine logische 0 oder eine logische 1 repräsentiert. Dieses Verfahren funktioniert unabhängig davon, wie unterschiedlich das GP vom durchschnittlichen Hintergrund ist. Als Beispiele für dieses Verfahren zeigen Figuren 5A, 5C, 5E und 5G vier verschiedene Zellen mit unterschiedlichen durchschnittlichen Pixelwerten vor dem Codieren einer logischen 0, und Figuren 5B, 5D, 5F und 5 H zeigen die vier Zellen von Figuren 5A, 5C, 5E und 5G,

20.02.01
DE 199 83 484 T 1

nachdem jeweils eine logische 0 codiert wurde. Figur 5I ist die Legende für Figuren 5A bis 5H. Jede Zelle hat ein Glyphpixel GP und Hintergrundpixel BP1, BP2, BP3, BP4, BP6, BP7, BP8, BP9. In Figuren 5A und 5C sind die Hintergrundpixel BP1 bis BP9 alle grau. In Figur 5A ist das GP vor dem Codieren weiß und in Figur 5C ist das GP vor dem Codieren schwarz; in beiden Fällen werden sie nach dem Codieren einer logischen 0 zu grau verändert (s. Figuren 5B und 5D). In Figur 5E sind einige der Hintergrundpixel schwarz und andere in verschiedenen Grautönen, die den durchschnittlichen Wert grau haben. Wie in Figur 5F dargestellt, wird das GP von einem vorcodierten schwarz auf einen postcodierten Grauwert gesetzt, um eine logische "0" zu codieren. In Figur 5G bestehen die durchschnittlichen Hintergrundpixel aus grauen, dunkelgrauen und schwarzen Pixeln, so daß der Durchschnitt dunkelgrau ist. Wie in Figur 5H dargestellt, wird das GP von hellgrau zu dunkelgrau geändert, um zum durchschnittlichen Hintergrund zu passen, um eine logische "0" zu codieren.

Aus den obigen Beispielen ist ersichtlich, daß eine Vielzahl von Hintergrundpixelwerten und Glyphpixelwerten in dem obigen Codierungsschema verwendet werden können. Die Pixel können beispielsweise jeden Grauton haben und jedes der Pixel in einer Zelle kann einen unterschiedlichen Grauton haben. So lange der durchschnittliche Hintergrundpixelwert bestimmt (berechnet) werden kann, kann die logische 1 durch Umkehren des Pixelwertes des GP vom durchschnittlichen Hintergrund codiert werden, während die logische "0" dadurch codiert werden kann, daß der Pixelwert des GP nahe am Hintergrund gehalten wird. Durch das Verwenden des obigen Verfahrens kann eine Nachricht in ein visuelles Bild eingebettet werden, in dem die eingebettete Nachricht weder auffällig noch täuschend bei einer visu-

20.02.01

DE 199 83 484 T1

ellen Inspektion ist, aber decodiert werden kann, indem das Verhältnis der Werte der Pixel in einem Bild analysiert wird.

Auch wenn in den obigen Ausführungsformen eine Zelle einen binären logischen Wert darstellt, ist es offensichtlich, daß "Superzellen" konstruiert werden können, in denen zwei oder mehr Glyphpixel mit benachbarten Hintergrundzellen in einer Zelle sind. Auf diese Art und Weise kann entweder mehr Information auf dem gleichen Raum dargestellt werden oder die gleiche Information kann robuster dargestellt werden. Mit Bezug auf Figur 3B kann die Zelle beispielsweise ein 5 x 5-Quadrat mit 25 Pixeln sein. Es gibt 4 GPs und 21BPs. Jedes der 4 GPs 1302, 1304, 1306 und 1308 kann einen separaten logischen Wert darstellen, die vier GPs können aber auch gemeinsam einen logischen Wert robuster darstellen.

Bild mit mehr als einer Farbe

Das Verfahren zum Codieren und Decodieren einer Nachricht in einem visuellen Bild kann in einem farbigen Bild ausgeführt werden, d.h. in einem Bild, das eine Mehrzahl von Farben enthält. Als Beispiel kann ein Bild, das aus vielfarbigen Pixeln (der Einfachheit halber manchmal als "Farbpixel" bezeichnet) besteht, die jeweils die drei Primärfarben (rot, blau, grün) haben, verwendet werden. Auf diese Art und Weise kann man eine ganze Palette von Farben erhalten, die die gewünschten Farben des Bildes bilden. Wenn dies gewünscht wird, können auch andere Farben als rot, blau und grün verwendet werden und das vorliegende Verfahren wird dennoch funktionieren, obwohl das visuelle Bild nicht so flexibel in der Darstellung einer Vielzahl von Farben sein kann. In der Ausführungsform der dreifarbigen Pixel (rot, blau, grün) kann jede Farbe, ähnlich wie bei

20.02.01
DE 199 83 484T1

den Graustufen, beim Pixelwert eine Helligkeitsskala von 0 bis 255, haben.

Ähnlich wie bei den Bildern in verschiedenen Graustufen kann eine Codierschwelle definiert werden als Mittel aus den hellsten und den dunkelsten Pixelwerten. Ebenfalls ähnlich zu der Ausführungsform in verschiedenen Graustufen, können die logischen Bits 0 und 1 codiert werden, indem die Helligkeit des Glyphpixels auf der Grundlage der durchschnittlichen Pixelwerte der Pixel, die das Glyphpixel umgeben, bestimmt wird. Bei drei Farben kann man, wenn dies gewünscht wird, dreimal soviel Information codieren, wie mit einem graustufigen Modell und zwar auf eine Art und Weise, die analog der "Superzelle" ist, wie oben für die Ausführungsformen in Graustufen beschrieben. Z.B. kann die rote Farbe verwendet werden, um den ersten Teil der einzubettenden Nachricht zu codieren, die grüne Farbe kann verwendet werden, um den zweiten Teil der Nachricht zu codieren und die blaue Farbe kann verwendet werden, um den dritten Teil der einzubettenden Nachricht zu codieren. Beim Decodieren können die drei Teile miteinander verbunden werden, um eine längere Gesamtnachricht zu bilden.

Um den Bitwert der logischen 1 zu codieren, nimmt das Glyphpixel kurz gesagt einen Pixelwert an, der entgegengesetzt zum durchschnittlichen Hintergrund ist (d.h. der den höchsten Kontrast zum durchschnittlichen Hintergrund bildet). Es nimmt also mit anderen Worten einen Pixelwert an, der vorzugsweise ganz unterschiedlich vom Pixelwert des durchschnittlichen Hintergrunds ist. Bei dieser "Umkehrung", wird für den roten Kanal der Pixelwert (pixel value, PV) auf 0 gesetzt, wenn der Hintergrundsdurchschnitt (background average, BA) größer als 127 oder gleich 127 ist; der PV wird auf 255 gesetzt, wenn der BA kleiner als 127 ist. Ähnlich wird auch beim blauen Kanal

20.02.01
DE 199 83 484 T1

und beim grünen Kanal der PV jeweils auf 0 gesetzt, wenn der Hintergrunddurchschnitt BA größer als 127 oder gleich 127 ist; der PV ist 255, wenn der BA kleiner als 127 ist. Wenn sich das Glyphpixel also im Verhältnis zum Hintergrund den Pixelwert "umdreht", um eine logische 1 zu codieren, dann bekommt es den extremen Pixelwert (d.h. das Pixel ist entweder das hellste 255 oder das dunkelste 0). Auf diese Art und Weise ist der Kontrast des Pixelwertes PV vor und nach dem Codieren am größten. Unter Verwendung dieses Schemas kann das Datenbit codiert werden. Dazu können bei drei unterschiedlich gesteuerten Farben bis zu drei Bitwerte in einem Farbpixel (color pixel, CP) transportiert werden, das wie hier verwendet ein Pixel von jeder Farbkomponente enthält (in dieser Ausführungsform rot, grün, blau).

Zur Illustration zeigen Figuren 6A bis 6H jeweils eine farbige Zelle, die die farbigen Hintergrundpixel BP1, BP2, BP3, BP4, BP6, BP7, BP8, BP9 hat, die ein zentrales farbiges Glyphpixel GP umgeben. Figur 6I ist die Legende für Figuren 6A bis 6H. Analog zu den einfarbigen Zellen bedeuten die Begriffe Glyphpixel GP und Hintergrundpixel (BP) in Bezug auf Farbpixel und Farbzellen entsprechend farbiges Glyphpixel und farbiges Hintergrundpixel. In jedem Farbpixel ist eine rote Komponente, eine blaue Komponente und eine grüne Komponente. In Figur 6A sind alle Farbpixel gelb und haben einen PV von (255, 255, 0), was darstellt, daß die rote Komponente einen Pixelwert von 255, die grüne Komponente einen PV von 255 und die blaue Komponente einen PV von 0 hat. Um eine logische 1 zu codieren, dreht das GP den PV des Hintergrunds um. Da die rote Komponente des Hintergrunds einen PV von 255 hat, hat das GP eine rote Komponente mit dem PV 0. Da die grüne Komponente des Hintergrunds einen PV von 255 hat, hat das GP eine grüne Komponente mit dem PV 0. Da die blaue Komponente des Hintergrunds

20.02.01
DE 199 83 484 T 1

einen PV von 0 hat, hat das GP eine blaue Komponente mit dem PV 255. So hat das GP nach dem Codieren einer logischen 1 einen PV von (0, 0, 255), was blau ist (s. Figur 6B).

Dieses Codierschema kann auf andere Pixelwerte (PV) ausgedehnt werden. Figur 6C zeigt eine Zelle, in der der PV von allen Pixel (0, 255, 255) ist, d.h. zyan (was hier tief grün und blau bedeutet). Da der Hintergrunddurchschnitt (0, 255, 255) ist, hat das GP nach der Umkehrung, um eine logische 1 zu codieren, einen PV von (255, 0, 0), was eine tiefrote Farbe ist (s. Figur 6D).

Um einen Datenbitwert der logischen 0 zu codieren, nehmen die drei Farbkomponenten jeweils den Pixelwert des durchschnittlichen Hintergrunds an. So sind beispielsweise in Figur 6E die Hintergrundpixel BP1 bis BP4 alle blau und haben einen PV von (0, 0, 255) und die anderen Hintergrundpixel BP 6 bis BP9 sind rot (255, 0, 0). Um eine logische 0 zu codieren, nimmt das GP die Farbe magenta an, d.h. einen PV von (127, 0, 127). Ähnlich zeigen Figur 6G (vor dem Codieren) und Figur 6H (nach dem Codieren) einen GP, dessen Hintergrund halb grün halb rot ist und der die Farbe gelb annimmt (das aus rot und grün zusammengesetzt ist), um eine logische "0" zu codieren.

Auch wenn dies hier nicht im Detail beschrieben wird, können andere Kombination von Farben mit verschiedenen Pixelwerten vom Fachmann auf der Grundlage der vorliegenden Offenbarung verwendet werden. In einem weiteren Beispiel (nicht in den Figuren dargestellt) ist das Glyphpixel grün (0, 255, 0). Die Hintergrundpixel beinhalten eine Vielzahl von Farben: Pixel BP1 ist gelb, Pixel BP2 und BP3 sind rot, Pixel BP4 und BP6 sind grün, Pixel BP7 und BP8 sind tiefblau, Pixel BP9 ist rot, wodurch ein durchschnittlicher Hintergrundpixelwert von (96,

20.02.01
DE 199 83 484 T 1

80, 96) entsteht. Folglich hat das Glyphpixel nach dem Codieren der logischen "0" einen Pixelwert von (96, 80, 96), was grau ist.

Codieren eines Glyphblocks

Das Schema des Codierens der logischen Werte "1" und "0" kann verwendet werden, um ein Bild aus Pixeln mit einer eingebetteten Nachricht zu codieren. Z.B. kann es gewünscht sein, eine URL-Adresse (Website-Adresse) in ein Bild einzubetten, und zwar so daß das Bild vom Betrachter leicht bemerkt und verstanden werden kann und ohne daß es zu einer gravierenden Störung durch die Nachricht kommt. Als spezifischeres Beispiel kann es z.B. gewünscht sein, die URL Adresse `http:\\www.webstar.com\\` in ein dargestelltes Bild, das das Wort "Webstar" darstellt, einzubetten. Figur 7 ist ein Bild aus Pixeln, die das Wort "Webstar" darstellen (d.h. das Vordergrundbild) ohne die eingebettete Nachricht. Nach dem Einbetten der Nachricht gemäß einem erfindungsgemäßen Verfahren, wird das Bild so, wie in Figur 8A dargestellt. Die Bilder in Figuren 7 und 8A sind im wesentlichen die gleichen unter dem Gesichtspunkt, daß das Vordergrundbild, das Wort "Webstar" in Figur 8A immer noch gut bemerkt werden kann. Aus der Entfernung sehen Figuren 7 und 8A gleich aus, wenn man von Grauübertönungen in Figur 8A absieht. Das Vorhandensein der codierten Nachricht wird durch das Vorhandensein von Punkten, d.h. schwarzen Pixeln im weißen Bereich und weißen Pixeln im schwarzen Bereich angezeigt.

Wenn man das gewünschte visuelle Bild und die in das visuelle Bild einzubettende Nachricht kennt, kann man einen Codierprozeß, der folgende Schritte beinhaltet, durchführen. So wie der Begriff hier verwendet wird, ist ein "Glyphblock" ein Block, d.h. eine Anordnung von Pixeln, die das Bild für das visuelle

20.02.01
DE 199 83 484 T1

Erkennen des Betrachters und die codierte Nachricht beinhaltet. So besteht der Glyphblock (im nachfolgenden der Einfachheit halber manchmal nur "Block" genannt) aus 1) einem benutzerdefinierten Vordergrundstring, der das visuelle Bild transportiert und 2) einem codierten Datenstrom, der eine Darstellung der 0en und 1en der Nachricht ist. Die Pixel, die die codierte Nachricht darstellen, sind nicht störend und werden bei flüchtiger Betrachtung durch einen Betrachter bei normalem Licht nicht bemerkt. Die Größe des Blocks wird durch die Größe des Vordergrundstrings definiert oder gemäß den Wünschen des Benutzers, er muß aber größer sein, als die Größe des Vordergrundstrings.

Für das nachfolgende illustrative Beispiel werden folgende Abmessungen verwendet:

Zellgröße = 3 x 3 Pixel;
Kachelgröße = 2 x 2 Zellen; und
Blockgröße = m x r Kacheln.

Der Glyphblock kann durch Ausführen folgender exemplarischer Schritte codiert werden:

1. Wähle die Glyphblockgröße aus. Auf der Grundlage des Vordergrundbildes, das zur visuellen Wahrnehmung gezeigt werden soll, weiß man die Vordergrundstringgröße, von der abgeleitet die Abmessungen des Glyphblocks ausgewählt werden können. Die Glyphblockabmessungen können so ausgewählt werden, daß das visuelle Bild, z.B. **"Webstar"** zur visuellen Betrachtung entsprechend dargestellt werden kann und die Nachricht eingebettet werden kann. Wie in Figur 7 dargestellt, kann das visuelle Bild **"Webstar"** in der Schrift "Times New Roman" in der Größe 48 Punkt zur visuellen Betrachtungen in einem Glyphblock mit 210 x 78 Pixeln dargestellt werden, d.h. 70 Spalten und 26

20.02.01
DE 199 83 48411

Reichen mit insgesamt 1820 Zellen, von denen jede 9 Pixel enthält. In einem Schema, in dem drei Datenzellen und eine Synczelle in einer Kachel sind, ist die gesamte Anzahl der Blockspalten (in der horizontalen Dimension gerechnet) gleich: 210 Pixels = 70 Zellen = 35 Kacheln. Die Gesamtzahl der Reihen (in der vertikalen Dimension berechnet) ist gleich: 78 Pixel = 26 Zellen = 13 Kacheln. Der Block enthält also insgesamt $70 \times 26 = 1820$ Zellen. In einer beispielhaften Kachelstruktur, die nur drei Datenbits pro Syncbit enthält, ist die Datenkapazität des Blocks zum Transportieren von Dateninformation deshalb $1820 \times (3/4)$ Bits, d.h. 1365 Bits, das sind 170 Bytes. Die Synchronisationskapazität des Blocks zum Speichern von Parametern, die Daten decodieren, ist $1820 \times (1/4)$ Bits, d.h. 455 Bits, das sind 56 Bytes.

2. Bestimme die einzubettende Nachricht. Abhängig von der Art der Nachricht, die der Benutzer codieren will, ist die Nachricht so lange richtig ausgewählt, wie die codierte Nachrichtengröße ein Teil der Größe des Glyphblocks ist, um eine entsprechende Fehlerkorrektur zu ermöglichen. Die Ebene der Fehlerkorrektur hat Einfluß darauf, wie groß oder klein die Nachricht sein kann. So ist z.B. die Glyphblockgröße mit einer Datenkapazität von 170 Byte ausreichend, um die URL "<http://www.webstar.com>", die eine Größe von 23 Bytes hat, zu codieren.

3. Erstelle die Blockfahne. Eine Blockfahne enthält für das Decodieren verwendete Informationen. Insbesondere enthält die Blockfahne Information über den Blocktyp, Block-ID, Nachrichtengröße und Fehlerkorrekturverfahren. Die Information über den Blocktyp und die Block-ID wird verwendet, um in Reihe geschaltete oder parallele Blöcke zu identifizieren. Die Information über die Nachrichtengröße und Fehlerkorrekturverfahren

20.02.01

DE 199 83 484 T 1

werden beim Fehlerkorrekturdecodieren verwendet. In einer Ausführungsform des Decodierens hat die Blockfahne eine feste Größe, nämlich 4 Byte, da die Blockfahne eine festgelegte Menge an Information enthält. In der obigen "webstar" Ausführungsform bedeutet dies:

Fehlerkorrekturgröße für Daten = Datenkapazität - Nachrichten-
größe = (170 - 23) Bytes

Fehlerkorrekturgröße für die Synchronisation = Synchronisati-
onskapazität/2 - Blockfahnenlänge = (56/2 - 4) Bytes.

Die Fehlerkorrekturgröße für die Daten bestimmt, wie viele Fehler im Datenabschnitt vorkommen dürfen, die noch toleriert werden können. Wenn beispielsweise der Reed-Solomon-Fehlerkorrekturcode verwendet wird und die Korrekturgröße für die Daten 157 Bytes ist, können bis zu 157 Fehlerbytes im Datenabschnitt sein und die korrekten Daten sind immer noch wiederherstellbar. Genauso können, wenn die Fehlerkorrekturgröße für die Synchronisation 22 Bytes ist, bis zu 22 Fehlerbytes im Synchronisationsbereich sein und die korrekte Synchronisation läßt sich trotzdem wiederherstellen.

4. Erstelle den Synchronisationsstrom (Syncstrom) durch Kombinieren der Blockfahne und der assoziierten Fehlerkorrekturdaten. Der Syncstrom ist ein Strom aus logischen 0en und 1en, der als Funktion des Blocktags generiert werden kann. Um die Blockfahne vor möglichen Fehlern zu schützen, kann ein Fehlerkorrekturverfahren, z.B. der allgemein bekannte Reed-Solomon-Fehlerkorrekturcode oder ein für die Funktion speziell entwickeltes Fehlerkorrekturverfahren verwendet werden:

Syncstrom = Blockfahne + Fehlerkorrekturfunktion (Blockfahne)

20.02.01

DE 199 83 484 T1

Verfahren zum Erstellen und Verwenden eines Fehlerkorrekturcodes in selbsttaktenden Glyphcodes sind im Stand der Technik bekannt, z.B. US-Patent U.S.P.N. 5,771,245 (Zhang), das hiermit durch Verweis aufgenommen wird. In einem bevorzugten Modus gemäß der vorliegenden Erfindung sind die Syncbits im Syncstrom so angeordnet, daß der Syncstrom symmetrische Charakteristika hat. Der Syncstrom hat beispielsweise eine Vorwärts- und eine Rückwärtskomponente, wobei die Vorwärts- und die Rückwärtskomponente gegengleich sind. Das Schema wird im Datenblockzellenlayout in Figur 13 dargestellt. Die Position der Synczellen wird durch s1, s2, s3 usw. angezeigt. Die Datenzellen werden durch Etiketten d1, d2, d3 usw. dargestellt. In dem dargestellten Beispiel hat der Block 1310 13 x 5 Kacheln, von denen jede 2 x 2 Zellen hat. Der Block 1310 hat also 32 Vorwärtssynchronisationszellen, die mit s1 bis s32 bezeichnet werden sowie 32 Rückwärtssynchronisationszellen, die mit s1' bis s32' bezeichnet werden. Die Zelle s1 (1301) im Vorwärtssynchronisationsstrom befindet sich in der ersten Kachel in der oberen linken Ecke von Block 1310. Die Zelle s1' (1302) im Rückwärtssynchronisationsstrom befindet sich in der letzten Kachel in der unteren rechten Ecke von Block 1310. Die Datenzellen sind einer Reihenfolge angeordnet, die bei der ersten Kachel in der unteren rechten Ecke beginnt (die die Datenzellen d1, d2, d3 enthält) bis zur letzten Kachel (die die Datenzellen d191, d192, p enthält, wobei p eine Füllzelle ist enthält). Alternative Schemata für symmetrische Verfahren können erreicht werden; es kann beispielsweise mehr als ein passender Syncstromabschnitt verwendet werden, indem vier Syncströme in den Ecken des Blocks angeordnet werden. Andere Symmetrieausführungsformen werden im nachfolgenden beschrieben, s. "In sich symmetrische Codes".

20.02.01

DE 199 83 484 T 1

5. Erstelle den Datenstrom durch Kombinieren des eingebetteten Nachrichtenstroms und des Fehlerkorrekturstroms. Ein Verfahren ist es, den eingebetteten Nachrichtenstrom und den Fehlerkorrekturstrom räumlich zu konzentrieren, (d.h. Datenstrom = Nachrichtenstrom + Fehlerkorrekturstrom). Dadurch, daß der in Figur 13 dargestellte Block beispielsweise das Konzentrationsverfahren verwendet, sind die ersten Datenbits in dem Nachrichtenstrom. Der Fehlerkorrekturstrom folgt unmittelbar auf den Nachrichtenstrom. Dem Fachmann wird klar sein, daß ein solches Konzentrationsverfahren nur den Zwecken der Darstellung dient und daß verschiedene Anordnungen des Nachrichtenstroms und des Fehlerkorrekturstroms verwendet werden können. Die Daten des Nachrichtenstroms und des Fehlerkorrekturstroms können sich vermengen, solange das Vermengungsverfahren beim Lesen des Bildes interpretiert werden kann, um die Nachricht zu dechiffrieren.

6. Erstelle den Datenblock durch Verschachteln des Syncstroms mit dem Datenstrom. Fülle jede Kachel zunächst mit einem Syncbit und den Rest der Kachel mit $(n \times n - 1)$ Datenbits. Die Position und der Inhalt des Syncbits sind bekannt und sind wie oben beschrieben ausgelegt. Wenn die Datenbits bekannt sind, können, da die Position der Syncbits ebenfalls bekannt ist, die Datenstrombits eingefügt werden, und zwar in einer Anordnung, in der ein Bit nach dem anderen angeordnet wird und zwar von oben nach unten und von links nach rechts, von einer Kachel zu nächsten, von der ersten Kachel zu letzten. Auf diese Art und Weise werden die Datenbits d1, d2, d3, d4 usw., wie in Figur 13 dargestellt, angeordnet. Das Beispiel einer Kachel 1312 mit vier Zellen (z.B. vier Zellen s1, d1, d2, d3) wird in Figur 13 dargestellt. Auf diese Art und Weise werden die Syncbits und die Datenbits ganz gleichmäßig über den Block ver-

20.02.01
DE 199 83 4 84 T 1

teilt. Dies stellt eine nicht störende Art und Weise dar, die Daten räumlich zu synchronisieren, was ein visuelles Bild ermöglicht, das im wesentlichen homogen ist, sei es ein Wort, ein Satz, ein Bild einer Obstschale.

7. Codiere den Datenblock von logischen Werten zu dem gewünschten visuellen Bild, um ein fertig codiertes Glyphbild, das Pixel enthält, herzustellen. Das Glyphbild zeigt ein Bild, das durch flüchtiges Betrachten erkennbar ist. S. z.B. Figur 8A, die den Glyphblock mit der eingebetteten Nachricht in einem visuell erkennbaren Bild "Webstar" darstellt.

Eine beispielhafte Ausführungsform für einen Nachrichtencodierprozeß, der verwendet werden kann, um eine eingebettete Nachricht zu codieren, wird in dem Ablaufdiagramm in Figur 9A dargestellt. Zunächst werden das darzustellende visuelle Vordergrundbild und die im Vordergrundbild einzubettende Nachricht bestimmt. Dies kann dadurch ausgeführt werden, daß eine einzubettende Nachricht ausgewählt wird (Block 902) und daß der Vordergrundstring bestimmt wird (Block 904). Dann wird die Nachricht in den Vordergrundstring codiert, um ein Glyphbild mit einer eingebetteten Nachricht herzustellen (Block 906). Das Glyphbild wird dargestellt, z.B. durch Drucken auf ein Medium, wie z.B. Papier (Block 908).

Detaillierter wird das Codieren einer beispielhaften Ausführungsform, die ein Vordergrundbild aus Worten darstellt, in Figur 9B dargestellt. Da die Worte und die Schriftart des Bildes des Vordergrundstring 912 bekannt sind, wird ein visuelles Bild 914 erstellt. Auf der Grundlage des visuellen Bildes werden die Blockparameter 916, wie die Blockabmessungen, bestimmt. Da auf der anderen Seite die einzubettende Nachricht 918 bekannt ist, werden auf der Basis der Blockparameter und

des gewählten Fehlerkorrekturverfahrens der Datenstrom und der Syncstrom 920 erstellt. Durch das Anordnen des Datenstroms und des Syncstroms 920 gemäß der Blockparameter 916 entsteht ein Glyphdatenblock (Datenblock) 922. Durch das Einfügen der logischen Werte des Datenblocks in das visuelle Bild entsteht ein Glyphbild, in das die Benutzerinformation eingebettet ist 924.

In sich symmetrische Codes

Wie oben dargestellt, ist es ein Verfahren, die Datenbits zu synchronisieren, einen Syncstrom in den Glyphblock mit dem Datenstrom einzuschließen, wobei der Syncstrom symmetrisch ist, während der Datenstrom asymmetrisch ist. Ein Vorteil der Symmetrie ist, daß sie eine Extraschicht für die Fehlerkorrektur bereitstellt. Ein anderer Vorteil ist, daß es möglich ist, die Blockposition und Abmessung schnell wiederherzustellen.

Figur 10 stellt eine Ausführungsform eines solchen Verfahrens dar. In Figur 10 enthält der in sich symmetrische Code Glyphzellen, die symmetrisch um eine globale Zentrumszelle 1000 gelegt werden. Die Glyphzelle 1001 (mit den Koordinaten x_1, y_1) und die Glyphzelle 1002 (mit den Koordinaten $-x_1, -y_1$) auf der gegenüberliegenden Seite der Zentrumszelle 1000 (mit den Koordinaten 0,0) bilden einen Zellenpaar, das sich spiegelt (um die Zentrumszelle 1000). In Zahlen ausgedrückt sind die Werte der beiden Zellen invers. Geometrisch gesehen sind die beiden Zellen diagonal symmetrisch um die Zentrumszelle 1000 angeordnet. Ein Paar von Glyphzellen codiert einen digitalen Bitwert. Wenn das primäre Bit 1001 größer als das Spiegelbit 1002 ist, ist der digitale Bitwert 1. Wenn das Gegenteil der Fall ist, hat das digitale Bit den Wert 0. Die Fehlersicherung ist offensichtlich, da das primäre Bit 1001 niemals mit dem Spiegelbit 1002 gleich sein kann. Wenn sie gleich

20.00.01
DE 199 83 484T1

sind, dann ist entweder in Zelle 1001 oder in Zelle 1002 ein Fehler. Die Möglichkeit, die Position und Abmessung eines Blocks schnell zu finden, ist ebenfalls offensichtlich, da alle symmetrischen Zellen um die Zentrumszelle 1000 angeordnet sind und die Symmetrie leicht getestet werden kann.

Im allgemeinen gibt es drei Typen von Blocklayouts für in sich symmetrische Codes: 1) erstes Layout: ein Block mit einer Schicht ohne Synczellen; 2) zweites Layout: ein Block mit zwei Schichten mit symmetrischen Synczellen und symmetrischen Datenzellen; und 3) drittes Layout: ein Block mit zwei Schichten mit symmetrischen Synczellen und asymmetrischen Datenzellen. Die Synczellen werden verwendet, um einen Speicherplatz für die Blockfahneninformation bereitzustellen und um weiterhin den assoziierten Block zu synchronisieren. Die Datenzellen werden dazu verwendet, um die Nachrichtendaten und die relevanten Fehlerkorrekturdaten zu speichern. Jeder doppelschichtige Block hat eine assoziierte Blockfahne, die, wie bereits erwähnt, Informationen wie die Nachrichtengröße, Fehlerkorrekturverfahren, Blockindex und Blocktyp enthält. Die Blockfahnen werden in den Syncstrom durch Fehlerkorrekturcodieren codiert. Typischerweise wird der erste Typ (ohne Synczelle) verwendet, um eine kleine Menge einer Benutzernachricht in ein kleines Bild zu codieren. In diesem Fall können anstelle der Synczelle, um die Blockfahne zu speichern, ein oder mehrere Defaultblockfahnen vorgesehen sein. In einer Defaultblockfahne entspricht die Fehlerkorrekturgröße beispielsweise der Hälfte der Datenkapazität des Blocks und in einer anderen Defaultfahne entspricht die Fehlerkorrekturfahne einem Viertel der Datenkapazität des Blocks usw. Während des Decodierens kann die Versuchs- und Irrtumsmethode verwendet werden. Wenn das Decodieren des einschichtigen Blocks unter Verwendung der ersten Defaultblockfahne nicht erfolgreich ist, kann die zweite

20.02.01

DE 199 83 484 T1

Defaultfahne, die dritte Defaultfahne usw. versucht werden.
Wenn das Decodieren unter Verwendung aller Defaultfahnen nicht funktioniert, kann der Block nicht decodiert werden.

Figur 11 zeigt eine Ausführungsform des ersten Layout: einen Block mit einer Schicht. In der Datenkarte (data map) 1102 haben die primären Datenzellen d1 bis d87 Spiegelzellen d1' bis d87', die eine zentralpositionierte Zentrumszelle (als "x" dargestellt) umgeben. Die primären Datenzellen sind diagonal symmetrisch mit Bezug auf die Spiegelzellen um die Zentrumszelle 1100. Die Synchronisation wird durch symmetrisches Anordnen der Datenzellen um die zentrale Zelle erreicht. Dieses Zellenlayout kann durch einen Zellensymmetrietest identifiziert werden (d.h. jede Zelle hat eine Spiegelzelle). Wenn das Bild gelesen wird und eine solche Symmetrie gefunden wird, kann der Anfang des Datenstroms lokalisiert werden und der Datenstrom kann analysiert (d.h. decodiert) werden, um die eingebettete Nachricht aufzudecken. In solchen selbsttaktenden Coden, sind die Synchronisationselemente (d.h. Elemente, die dem Leser die räumliche Synchronisation mitteilen, damit das Decodieren in der richtigen Reihenfolge gehalten werden kann) Datenbits, die Daten transportieren, die eingebettete Nachrichten oder relevante Fehlerkorrekturdaten enthalten können. Die Datenkarte 1102 hat 25 X 7 Datenzellen. Die Datenkarte hat einen Vorwärtsdatenstrom 1104 (repräsentiert durch die Zellen d1, d2, d3, ... d87) und einen Rückwärtsdatenstrom 1106 (dargestellt durch die Zellen d1', d2', d3', ... d87'). Die Bits des Vorwärtsdatenstroms 1104 beginnen bei der ersten Zelle (d1), da sich die Bits nach vorne bewegen und die Bits des Rückwärtsdatenstroms 1106 beginnen, mit ihrem "ersten" Bit an der letzten Zelle (d1') und bewegen sich rückwärts in die Datenkarte 1102. Durch Vergleichen der Bits, die an einer Ecke

20.02.01
DE 199 83 484 T1

beginnen mit den Bits, die an der anderen Ecke beginnen, kann die Orientierung des Datenstrom bestimmt werden und die Decodierung kann dann ordnungsgemäß durchgeführt werden, um die eingebettete Nachricht wiederherstellen. Da die Datenkarte diagonalsymmetrisch ist, kann ein Aufgreifen der Zentrumszelle x und ein Vergleichen der angrenzenden Zellen auch die Orientierung der Zellen zeigen. Wenn die Zelle $d87$ beispielsweise das Spiegelbild des logischen Wertes der Zelle $d87'$ ist und die Zelle $d86$ ein Spiegelbild der Zelle $d86'$ ist und so weiter, dann kann man daraus schließen, daß der Datenstrom bei Zelle $d1$ beginnt und den Decodierprozeß fortsetzten. Mit dem vorliegenden Verfahren können alle Bits vom ersten bis zum letzten Bit des Datenstroms korrekt bestimmt werden, ohne Verwendung von Syncbits, die nicht Teil des Datenstroms sind.

Der zweite Blocktyp mit symmetrischen Synczellen und symmetrischen Datenzellen wird verwendet, um eine mittlere Menge von Benutzernachrichten zu codieren. Der dritte Typ mit symmetrischen Synczellen und asymmetrischen Datenzellen wird verwendet, um eine größere Menge von Benutzerdaten zu codieren. Bei einem Block mit einer Schicht wird die Fehlerkorrektur nur am Nachrichtenstrom ausgeführt. (Zu beachten ist, daß der Datenstrom der Nachrichtenstrom plus Fehlerkorrekturstrom ist.) Bei einem doppelschichtigen Block wird die Fehlerkorrektur sowohl am Syncstrom als auch am Nachrichtenstrom ausgeführt. Wenn die Fehlerkorrektur am Syncstrom nicht erfolgreich ist, ist es nicht nötig, die Fehlerkorrektur am Nachrichtenstrom auszuführen. Da der Syncstrom viel kleiner als der Nachrichtenstrom ist, kann die assoziierte Fehlerkorrektur viel schneller versucht werden. Wenn der Syncstrom erfolgreich wiederhergestellt wurde, sind die entsprechende Blockfahne, Blockposition und Abmessung korrekt. Folglich muß die Fehlerkorrektur am Nachrichtenstrom nur einmal durchgeführt werden. Diese Eigenschaft

ermöglicht es, die Fehlerkorrektur am Syncstrom oft auszuführen, wenn man versucht, die richtige Blockfahne und die Blockposition herauszufinden, ohne die allgemeine Decodiergeschwindigkeit wesentlich zu verlangsamen.

Ein doppelschichtiger Block hat sowohl Synczellen als auch Datenzellen. Figur 12A zeigt beispielsweise einen doppelschichtigen Block 1210, der sowohl Syncsymmetrie als auch Datensymmetrie aufweist. Der Block ist kachelsymmetrisch, d.h. jede Kachel hat eine Spiegelkachel auf der entgegengesetzten Seite der Zentrums-kachel 1200. Die primäre Kachel 1201 und ihre Spiegelkachel 1202 sind diagonal symmetrisch um die Zentrums-kachel 1200. Ein anderes Beispiel für die Symmetrie zeigen Kacheln 1203 und 1204. Die Zellen s1 bis s32 zeigen einen Vorwärtssyncstrom, während die Zellen s1' bis s32' einen Rückwärtssyncstrom zeigen. Die Zellen d1 bis d92 zeigen einen Vorwärtsdatenstrom, während die Zellen d1' bis d92' einen Rückwärtsdatenstrom zeigen. Alle Zellen, die mit dem Buchstaben p bezeichnet sind, sind Füllzellen (padding cells), die leere Zellen besetzen, um eine Vollständigkeit der Kacheln zu ermöglichen. Figur 12B zeigt einen Vorwärtssyncstrom mit der Blockfahne und dem Fehlerkorrekturcode und Figur 12C zeigt einen Rückwärtssyncstrom, der ein Spiegelbild des Vorwärtssyncstroms aus Figur 12B ist. Figur 12D zeigt einen Vorwärtsdatenstrom mit dem Nachrichtenstrom und dem Fehlerkorrekturcode und Fig. 12F zeigt einen Rückwärtsdatenstrom, der ein Spiegelbild des Vorwärtsdatenstroms von Figur 12D ist.

Wie bereits erwähnt zeigt Figur 13 einen doppelschichtigen Block 1310 mit symmetrischen Synczellen und asymmetrischen Datenzellen um eine Zentrums-kachel 1300. Die Synczelle x der Zentrums-kachel 1300 ist die Zentrumssynczelle aller Synczellen. Die Zellen s1 bis s32 stellen den Vorwärtssyncstrom dar.

Die Zellen s1' bis s32' stellen den Rückwärtssyncstrom dar. Die Synczelle s1 (bezeichnet durch Zelle 1301) hat eine entsprechende Spiegelzelle s1' (bezeichnet durch Zelle 1302) und so weiter. Die Zellen d1 bis d92 stellen den Datenstrom dar und haben keine Spiegelzellen, da der Datenstrom asymmetrisch ist. Die Zelle 1304 ist eine Füllzelle.

Wenn dies gewünscht wird, kann ein Synchronisationsverfahren gewählt werden, bei dem sowohl der Syncstrom als auch der Datenstrom asymmetrisch sind. Auf der Grundlage der vorliegenden Offenbarung ist die Verwendung von asymmetrischen Synchronisationsmethoden für den Fachmann verständlich und wird hier nicht weiter im Detail beschrieben.

Decodieren des Glyphblocks

Wenn ein visuelles Bild mit einer eingebetteten Nachricht vorgegeben ist, decodiert man das dargestellte Bild, indem das dargestellte Bild als Glyphbild festgehalten wird und das Glyphbild transformiert wird, um die Bedeutung des Vordergrundbildes zu reduzieren und um die dargestellten logischen Werte zu finden.

Von dem Pixelwert einer Glyphzelle kann der logische Wert der Glyphzelle decodiert werden, indem das Codierverfahren von "1" und "0", wie oben beschrieben, umgekehrt wird.

Um das Decodieren des Glyphblocks zu implementieren, können folgende Schritte verwendet werden:

1. Halte das Glyphbild in einer Bildlesevorrichtung fest. Das Bild der "Webstar"-URL eingebettet in dem graustufigen Bild aus Figur 8A, kann beispielsweise unter Verwendung eines zwei-

dimensionalen CCD-Bildlesers oder CMOS-Bildlesers (einer Kamera usw.) gelesen werden.

2. Stelle durch Pixeltransformation die Glyphkarte her. Diese transformiert das Glyphbild (z.B. Figur 8A) in eine Glyphkarte (z.B. in Figur 8B dargestellt). In Figur 8B repräsentieren weiße Pixel in dieser Glyphkarte Glyphpixel mit der logischen 1. Schwarze Pixel können Glyphpixel mit dem logischen Wert 0 oder Hintergrundpixel sein. Um die Glyphkarte herzustellen, wird eine Pixeltransformation am Glyphbild durchgeführt, indem der transformierte Pixelwert aus den gemessenen Pixelwerten mit einer Gleichung berechnet wird.

Zu diesem Zweck kann die Pixeltransformation, um die Glyphkarte aus dem Glyphbild herzustellen, durchgeführt werden, indem zunächst der Wert des Zellkontrasts (cell contrast CC) für jedes Pixel berechnet wird, d.h. die absolute Differenz des Pixelwertes vom Durchschnitt der umgebenden Pixel (z.B. die absolute Differenz zwischen einem GP und seinen umgebenden Hintergrundpixeln BP). Der CC wird für alle Pixel, also GPs und BPs berechnet. In der Ausführungsform, in der das GP von allen vier Seiten von Hintergrundpixeln umgeben ist (z.B. in einer 3 x 3 Zelle wie in Figur 3A) ist der CC beispielsweise:

$$CC = ABS(GP_m - (BP1 + BP2 + BP3 + BP4 + BP6 + BP7 + BP8 + BP9)/8)$$

wobei ABS die absolute Funktion ist, GP_m der gemessene Pixelwert des Glyphpixels (GP) und BP1, BP2, BP3, BP4, BP6, BP7, BP8 und BP9 die gemessenen Pixelwerte der Hintergrundpixel sind, die den GP unmittelbar umgeben (s. Figur 3A). Dies resultiert in einer Glyphkarte, wie in Figur 8B dargestellt, die weiße Pixel, schwarze Pixel und graue Pixel enthält. Es ist zu beachten, daß alle Hin-

tergrundpixel im allgemeinen einen CC mit kleinem Wert haben und schwarz erscheinen. Aufgrund des verwendeten Codierverfahrens erscheinen die logischen Werte von "0" ebenfalls schwarz in Figur 8B. Weiße Pixel in Figur 8B repräsentieren den logischen Wert "1".

Es ist zu erwähnen, daß dieses Verfahren der Berechnung des Zellkontrasts für eine Vielzahl von Zellgrößen und Zellkonfigurationen verwendet werden kann, solange die relative Position zwischen einem GP und den BPs vordefiniert ist. In einer 2 x 2 Zelle, wenn die Pixel z.B. in einem regulären Muster (z.B. ist das GP in jeder Zelle in der oberen linken Ecke und die Hintergrundpixel BPs sind in den anderen Ecken des Quadrats) angeordnet sind, wird jedes GP immer noch von BPs umgeben, auch wenn einige der BPs in der obigen Gleichung von einer oder mehr Zellen für einen CC eines GP stammen können. Auf ähnliche Weise kann die Gleichung verwendet werden, um den CC für andere Zellgrößen zu berechnen.

Wenn der CC-Wert eines Glyphpixels GP über einem bestimmten vorbestimmten Wert ist (s. Figur 8B), dann ist es im wesentlichen so, daß das GP eine logische "1" ist. Wenn der CC für das GP klein ist, dann hat das GP den logischen Wert "0". Um also den logischen Wert von einem Glyphpixel zu bestimmen, wird der Wert eines GP mit einer Decodierschwelle (decoding threshold, DT) verglichen, die dem Wert von ET (die Codierschwelle - encoding threshold beim Codieren) entsprechen kann. Wenn die ET z.B. der durchschnittliche Pixelwert aus dem dunkelsten Pixel (Pixelwert 0) und dem hellsten Pixel (Pixelwert 255) ist, dann kann die DT der Durchschnitt zwischen dem hellsten Wert und dem dunkelsten Wert der Pixel in Figur 8B sein. Der logische Wert, d.h. der Bitwert (bit value, BV) eines Glyphpixel GP ist:

$$BV = 1 \text{ wenn } CC > DT \text{ oder } CC = DT$$

BV = 0 wenn CC < DT.

Diese Bitwerte (BV) bilden den normalisierten, transformierten, Pixelwert des GP und resultieren in einer Karte mit logischen "1" und "0" Positionen.

Dieses Decodierschema wird im folgenden Beispiel dargestellt. In einer Glyphzelle ähnlich der Zelle in Figur 4B ist beispielsweise eine schwarze, zentrale Zelle von weißen Hintergrundzellen BP1, BP2, BP3, BP4, BP6, BP7, BP8 und BP9 umgeben. Nach der Transformation ist das zentrale Pixel GP weiß und die Hintergrundpixel sind schwarz. Nach der Berechnung kann bestimmt werden, daß der BV 1 ist. Das ist genau der logische Wert, der mit dem oben beschriebenen Codierverfahren für Figur 4B codiert wurde. Ähnlich ist bei einer Glyphzelle wie in Figur 4D, eine weiße, zentrale Zelle GP von schwarzen Hintergrundzellen BP1, BP2, BP3, BP4, BP6, BP7, BP8 und BP9 umgeben. Nach der Transformation bleibt das weiße GP weiß, während die schwarzen GPs schwarz bleiben. In diesem Fall ist der CC ebenfalls größer als DT. So ist der BV ebenfalls 1. Ähnlich können die Glyphzellen, wie in Figuren 4F, 4H, 5B, 5D usw. dargestellt, transformiert werden und ihre CC-Werte mit der DT verglichen werden, um die logischen Werte zu bestimmen.

In Ausführungsformen mit mehrfarbigen Pixeln können die Pixel auf analoge Art und Weise transformiert werden und die logischen GP-Werte berechnet werden, indem die CC-Werte mit der DT für die verschiedenen Farben verglichen werden.

3. Erstelle die Zellkarte unter Verwendung eines zweidimensionalen Histogramms, um die Zellgröße zu bestimmen.

Um die Datenkarte (Figur 18) zu erstellen, in der alle logischen Werte der GPs bekannt sind, muß die Position der GPs genau bestimmt werden. Um dies durchzuführen, können eine Zellkarte, die die Zellabmessungen und die Zellkonfiguration (d.h. die Beziehung zwischen benachbarten Zellen) zeigt und eine Rasterkarte, die das Raster entlang dem die GP-Pixel ausgerichtet sind, zeigt, ermittelt werden. Das folgende Verfahren kann verwendet werden:

(A) Finde alle lokalen Maximumpixel in der Glyphkarte. Ein "Lokales Maximumpixel" ist ein Pixel, dessen Pixelwert größer oder gleich (\geq) dem Pixelwert der anliegenden Pixel in jeder Pixelmatrix ist, z.B. 3 x 3 Pixel, nicht auf eine einzige Zelle beschränkt. Wie im nachfolgenden noch detailliert beschrieben, werden lokale Maximumpixel verwendet, um die Zellabmessungen beim Decodieren der Information in einem Glyphblock zu bestimmen. Ein "Lokales Mittel" ist der durchschnittliche Pixelwert von jedem 3 x 3 Pixel. Ein lokales Mittel hat einen Wert zwischen 0 und 255 für die Pixel in Figur 8B. Ein Fachmann kann einen mathematischer Algorithmus formulieren und einfach ein Computerverfahren implementieren, um z.B. unter Verwendung der Pixelwerte von Figur 8B die Pixelwerte in der Glyphkarte zu vergleichen, um die lokalen Maximumpixel herauszufinden.

(B) Wähle eine lokale Fenstergröße $w \times w$ so, daß das Fenster groß genug ist, um eine Anzahl von Zellen zu bedecken. Es kann beispielsweise ein w -Wert von $\{w = 2 \times (\text{maximale Zellgröße}) + 1\}$ verwendet werden; wenn man davon ausgeht, daß die maximale Zellgröße 6 x 6 Pixel ist, dann ist $w = 2 \times 6 + 1 = 13$. Ein "lokales Fenster" ist ein Fenster von $w \times w$ Pixeln, wobei das lokale Maximumpixel im Zentrum jedes Fensters ist. Alle lokalen Maximumpixel sollten ein lokales Fenster haben.

(C) Erstelle ein zweidimensionales Histogramm der Größe $w \times w$, d.h. ein 13×13 Histogramm durch Stapeln (d.h. Summieren) aller Pixelwerte aller lokalen Fenster des Histogramms. Der Einfachheit halber können die normalisierten, transformierten Pixelwerte "0" und "1" verwendet werden. Es ist offensichtlich, daß die nicht-normalisierten Pixelwerte aus Figur 8B verwendet werden können, um das Histogramm zu bilden, und daß die Zellgröße immer noch durch Suchen nach den Maxima im Histogramm bestimmt werden kann.

(D) Nachdem alle Fensterwerte akkumuliert sind, wird das Histogramm zur Zellkarte (in Figur 8C), die die akkumulierten Pixelwerte um die Nachbarschaft der lokalen Maximumpixel zeigt. Statistisch stellen die großen Werte die Glyphpixel dar und die kleinen Werten stellen die Hintergrundpixel dar. Durch das Verwenden der Zellkarte als Schablone und ihr Ausdehnen über die Glyphkarte (z.B. Figur 8B) können die Positionen der Glyphpixel (GPs) ermittelt werden.

Zur Bestimmung der Zellgröße kann folgendes Verfahren verwendet werden.

(A) Finde das Zentrumspixel der akkumulierten lokalen Fenster in der Zellkarte (z.B. im Zellkartenabschnitt wie in Figur 8C dargestellt). Die höchste Zahl in der Karte stellt ein GP dar, z.B. 9 in der Zellkarte, wie in Figur 8C dargestellt. Finde auch vier andere lokale Maximumpixel, die dem Zentrum am nächsten sind (diese können linkes /left/, rechtes /right/, oberes /above/ und unteres /below/ Glyphpixel oder LGP, RGP, AGP und BGP genannt werden), die die herausragenden Zahlen in der Zellkarte sind (die herausragenden Glyphpixel sind alle 5en in der Ausführungsform von Figur 8C).

(B) Berechne die Zellgröße als den mittleren Abstand vom Zentrumspixel zu den vier benachbarten Glyphpixeln (GP). Die Zellgröße in Figur 8C ist beispielsweise $(3 + 3 + 3 + 3)/4 = 3$. D.h. die wiederhergestellte Zellabmessung ist 3×3 Pixel.

4. Erstelle die Rasterkarte unter Verwendung der Zellkarte und der Glyphkarte. Um die richtige Blockposition und Blockabmessung in einem gescannten Bild sicherzustellen, kann man ein Raster bestimmen, in dem alle GPs (Daten-GPs und Sync-GPs, soweit vorhanden) lokalisiert sind. Die Rasterkarte (s. Figur 8D) wird zu diesem Zweck erstellt. Da man die Zellgröße kennt, wird für jedes Pixel in der Glyphkarte, von dem angenommen wird, daß es ein GP-Pixel ist, der Pixelwert (p_i) nachgeprüft, indem es zu den vier nächsten angenommenen Pixel links (lp_i), rechts (rp_i), oben (tp_i) und unten (bp_i) addiert wird, um einen aktualisierten (oder revidierten) Rasterpixelwert zu bilden, d.h.

$$p_i, \text{ revidiert} = p_i + lp_i + rp_i + ap_i + bp_i.$$

Dieser Prozeß wird mehrmals wiederholt, z.B. 4 oder 5 mal. Bei einem GP-Pixel im Block wird, selbst wenn es mit einem niedrigen Pixelwert beginnt (z.B. bei einer logischen "0" Pixel mit geringem Kontrast zu den BPs), aufgrund der höheren Pixelwerte an den lokalen Maxima (d.h. Glyphpixel), nach ein paar Iterationen der Pixelwert progressiv steigen, da die benachbarten Pixel hell (d.h. mit hohen Pixelwerten) sind. Auf diese Art und Weise wird ein Raster mit hellen Pixeln geformt, um einen Block darzustellen, in dem die Glyphpixel lokalisiert sind. Dieses Verfahren stellt ein Äquivalent dar zu dem Untersuchen der Glyphkarte, um weiße sich abhebende Punkte (Pixel) zu ermitteln und ihr Füllen in die Glyphkarte in die schwarzen Ab-

schnitte zwischen die weißen Punkte an Stellen, die eine Zellenlänge entfernt sind, um alle Glyphpixelpositionen zu bestimmen. Das obige Berechnungsverfahren ist nur ein praktisches Verfahren, um die Position der Glyphpixel sicher festzustellen, egal ob sie in einem "0"- oder "1"-Status sind. Der Fachmann kann auch andere Verfahren auswählen.

5. Generiere die normalisierte logische Datenkarte so, daß die Kachelgröße des Glyphblocks gefunden und daß die Blockfahne wiederhergestellt werden kann. Um die normalisierte logische Datenkarte (z.B. Figur 18) zu erstellen, kann das folgende Verfahren verwendet werden. Man nimmt die Koordinaten aller lokaler Maximumpixel in der Glyphkarte (z.B. Figur 8D) und erhält die entsprechenden Pixelwerte von der Glyphkarte (s. z.B. Figur 8B). Dazu wird die Zellrepräsentation des Glyphblocks erstellt, d.h. durch das Herausnehmen der GPs, werden alle Hintergrundpixel fallengelassen. Auf diese Art und Weise kann man die Hintergrundpixel und die GPs mit logischer "0" in der transformierten Glyphkarte unterscheiden (s. Figur 8B), obwohl beide denselben Pixelwert in der Glyphkarte haben. Folglich kann die ganze Datenkarte auf einige vorbestimmte Minimum- und Maximumwerte normalisiert werden. Dann kann man bestimmen, ob der Block ein Einzelschicht- oder ein Doppelschichtblock ist, indem ein Symmetrietest durchgeführt wird. Bei einem doppelschichtigen Block ermittelt der Symmetrietest auch die Kachelgröße.

6. Stelle den Syncstrom und die darauf bezogene Blockfahne wieder her. Bei einem doppelschichtigen Block mit Syncsymmetrie wird der Syncstrom durch Bestimmen der Symmetrie wiederhergestellt. In einer Ausführungsform mit einer zentralen Kachel wird zunächst die zentrale Kachel definiert, da der Vorwärts- und der Rückwärtssyncstrom zirkelsymmetrisch um die

Zentrums-kachel verlaufen, und dann werden die Vorwärts- und Rückwärtsströme wiederhergestellt. Da der Vorwärts- und der Rückwärtsstrom zahlenmäßig invers zueinander sind, kann der binäre Wert (binary value, BV) der Syncbits wiederhergestellt werden, indem die Vorwärtssyncbits mit den Rückwärtssyncbits verglichen werden. Der BV von einem Paar von Syncbits ist eins, wenn der Wert des Vorwärtssyncbits größer als der des Rückwärtssyncbits ist. Der BV von einem Paar Syncbits ist null, wenn der Wert des Vorwärtssyncbits kleiner als der des Rückwärtssyncbits ist. Der BV von einem Paar von Syncbits ist unbekannt, wenn der Wert des Vorwärtssyncbits gleich dem des Rückwärtssyncbits ist, was bedeutet, daß ein Fehler in einem der Syncbits ist. Um die im Syncstrom eingebettete Blockfahne wiederherzustellen, benötigt man die relevante Fehlerkorrekturgröße. Designmäßig ist die Größe der Blockfahne festgelegt (vier Bytes); die Fehlerkorrekturgröße für den Syncstrom kann deshalb berechnet werden, indem die Blockfahnen-größe von der Syncstromgröße subtrahiert wird. Wenn man die Fehlerkorrekturgröße kennt und ein vorbestimmtes Fehlerkorrekturverfahren bekannt ist, kann die Fehlerkorrekturdecodierung durchgeführt werden. Wenn die Fehlerkorrekturdecodierung erfolgreich ist, sind die korrekte Stelle, Abmessung und Anordnung des relevanten Blocks bestimmt.

7. Stelle den Datenstrom und die eingebettete Benutzernachricht wieder her. Bei einem Block mit symmetrischen Datenzellen ist die Wiederherstellung des Bitwertes (bit value BV) des Datenstroms ähnlich der des Syncstroms. Bei einem Block mit asymmetrischen Datenzellen wird die Wiederherstellung des BV durch die folgenden Schritte erreicht.

(A) Berechne die lokale Schwelle (local threshold, LT). In einer Glyphkarte ähnlich der in Figur 8B muß beispielsweise

für jede Kachel, wenn die Synczelle = 1 ist, die nächste Kachel mit der Synczelle 0 gefunden werden; im anderen Fall (d.h. wenn Sync = 0), muß die nächste Kachel mit der Sync = 1 gefunden werden. Dann ist LT der durchschnittliche Pixelwert von Sync logisch 1 und Sync logisch 0.

(B). Decodiere den Bitwert BV der Datenzellen in einer Kachel.

BV = 1, wenn der Pixelwert der Datenzelle \geq LT

BV = 0, wenn der Pixelwert der Datenzelle $<$ LT

Es muß bemerkt werden, daß der Wert von LT auf der Basis der Schwelle, die für das Codieren der logischen 0 und logischen 1 im Codierprozeß verwendet wird, ausgewählt wird. Das Auswählen einer Schwelle als Mittel der Pixelwerte der logischen 0 und der logischen 1 ist nur eine Ausführungsform der vorliegenden Erfindung.

(C) Wiederhole die obigen Schritte in allen Kacheln. Es entsteht daraus eine Datenkarte, die die logischen Werte aller Bits hat (ähnlich der in Figur 18 dargestellten Karte). Die Datenkarte wird hergestellt, indem die Rasterkarte über die Glyphkarte abgebildet wird (mapping), um die logischen Werte zu erhalten. Es ist offensichtlich, daß der Fachmann die Rasterkarte über die Glyphkarte abbilden kann, um die logischen Werte für die Zellen zu erhalten, sogar um lineare und nicht-lineare Abweichung zu kompensieren. In der in Figur 18 dargestellten Ausführungsform wird eine in sich symmetrische, doppelagige, syncsymmetrische und datensymmetrische Karte gezeigt. Die zentrale Kachel 1801 wird durch 4 "x" dargestellt. Die eingebettete Nachricht wird wiederhergestellt, indem der decodierte Datenstrom verwendet wird. Es wird eine Fehlerkorrektur am Datenstrom ausgeführt, um die eingebettete Nachricht wiederherzustellen. Dies wird durchgeführt, indem die Funk-

tion, die für die Herstellung des Datenstroms von der eingebetteten Nachricht verwendet wurde, umgekehrt wird:

Eingebettete Nachricht = Funktion (Datenstrom)

Wenn die Fehlerkorrektur erfolgreich ist, kann die eingebettete Nachricht korrekt wiederhergestellt werden, z.B. Eingebettete Nachricht = **http:\\www.Webstar.com**.

Figur 19A illustriert in einem Ablaufdiagramm kurz den Prozeß des Decodierens einer eingebetteten Nachricht und des Antwortens auf diese Nachricht. Zum Decodieren wird ein Glyphbild zunächst erfaßt, z.B. mit einem zweidimensionalen Lesegerät (Block 1910). Das Glyphbild wird decodiert, um die eingebettete Nachricht wiederherzustellen (Block 1912). Um zu antworten kann ein Prozessor auf die decodierte, eingebettete Nachricht hin in Aktion treten, z.B. indem ein Webbrowser (z.B. den Microsoft Internet Explorer, Microsoft Corporation, den Netscape Communicator, Netscape Corp. usw.) so gesteuert wird, daß er mit der URL-Adresse, die die eingebettete Nachricht ist, verbindet.

Figur 19B zeigt detaillierter eine Ausführungsform des Decodierverfahrens. Zunächst wird das Glyphbild 1922 erfaßt. Eine Glyphkarte 1924 wird durch Pixeltransformation erstellt. Eine Zellkarte 1926 wird von der Glyphkarte hergestellt. Eine Rasterkarte 1928 wird auf der Basis der Glyphkarte 1924 und der Zellkarte 1926 hergestellt. Eine Hauptfunktion einer Glyphkarte ist es, die Glyphzellen herauszufiltern, d.h. sie zu identifizieren. Die Hauptfunktion der Zellkarte ist es, die Abmessungen und Orientierung wiederherzustellen. Eine Hauptfunktion der Rasterkarte ist es, Ankerpunkte für die Zel-

len bereitzustellen. Die Ankerpunkte sind wichtig, da alle Zellen mit dem logischen Wert 1 in der Glyphkarte offensichtlich sind. Der Glyphdatenblock (Datenblock) 1930 wird durch Abbilden (mapping) der Glyphkarte hergestellt, wobei die Zellekarte 1926 und die Rasterkarte 1928 als Führung vom Bildbereich zum Datenbereich verwendet wird. Der Syncstrom 1932 wird wiederhergestellt, indem nach der Symmetrie und spezifischen Anordnungen gesucht wird nach verschiedenen Schemen vom Glyphdatenblock 1930. Durch das Ausführen der assoziierten Fehlerkorrekturdecodierung werden die Nachrichtengröße und die Blockparameter 1936 von der Blockfahne, die in dem Syncstrom eingebettet ist, wiederhergestellt. Die erfolgreiche Wiederherstellung des Syncstroms sichert die korrekten Positionen und Abmessungen des Datenstroms 1934. Die Benutzernachricht wird decodiert, wenn die Nachrichtengröße und die Blockparameter verfügbar sind, um den Datenstrom zu interpretieren.

Was farbige Pixel betrifft, kann für das Decodieren von Glyphzellen in farbigen Bildern ein ähnliches Schema gewählt werden, wie für das Decodieren von Glyphzellen in Graustufen im einfarbigen Verfahren. Kurz gesagt umfaßt das Verfahren das Herausfinden, ob eine Glyphzelle einen hohen Kontrast mit den Hintergrundpixeln hat (d.h. ob das Glyphpixel einen Wert hat, der sich von dem des durchschnittlichen Hintergrunds um mehr als die Hälfte des maximalen Pixelwertes unterscheidet). Bei diesem Decodierschema wird der absolute Wertunterschied des Pixelwertes zwischen dem durchschnittlichen Hintergrund und dem farbigen Glyphpixel GP berechnet und "Zellkontrast" (cell contrast "CC") genannt. Zum Decodieren wird der maximale Wert der drei Farbkomponenten von CC ermittelt. Beim Decodieren ist der logische Bitwert 1, wenn der maximale (CC) gleich oder größer ist als 127, was das Mittel aus dem hellsten und dem

dunkelsten Pixel in einer Farbe ist. Der logische Bitwert ist 0, wenn der maximale (CC) kleiner als 127 ist.

Als exemplarische Beispiele zeigen Figuren 6B und 6D wie farbige Zellen decodiert werden können. In Figur 6B ist GP ein blaues Pixel mit einem Pixelwert von (0,0,255) mit Bezug auf die roten, grünen und blauen Komponenten. Die Hintergrundpixel BP1, BP2, BP3, BP4, BP6, BP7, BP8 und BP9 sind gelb und haben jeweils einen Pixelwert von (255, 255, 0). Das durchschnittliche Hintergrundpixel hat deshalb einen Pixelwert von (255, 255, 0). CC hat deshalb einen Pixelwert von (255, 255, 255). Folglich ist der maximale (CC) 255, was anzeigt, daß der logische Wert von GP 1 ist. In Figur 6D ist das GP ein rotes Pixel und hat einen Pixelwert von (255, 0,0) mit Bezug auf die roten, grünen und blauen Komponenten. Die Hintergrundpixel BP1, BP2, BP3, BP4, BP6, BP7, BP8 und BP9 sind zyan (d.h. blaugrün) und haben jeweils einen Pixelwert von (0, 255, 255). Der CC hat deshalb einen Wert von (255, 255, 255). Der maximale (CC) ist also 255, was anzeigt, daß der logische Wert des GP 1 ist. Andere farbige Glyphzellen können auf analoge Art und Weise decodiert werden. Für den Fachmann ist es verständlich, daß solche Verfahren des Codierens und Decodierens der logischen 1 und der logischen 0 unter Verwendung von Farben und die anderen Merkmale zum Einbetten einer Nachricht in ein visuelles Vordergrundbild und zum Wiederherstellen der eingebetteten Nachricht implementiert werden können, mit Verfahren, die ähnlich denen der graustufigen Ausführungsform wie oben beschrieben sind.

Visuelle Bilder in mehreren Blöcken

Die vorliegende Erfindung kann verwendet werden, um visuelle Bilder zu codieren und decodieren, die sich über zwei oder

20.00.01
DE 199 83 484 T 1

mehr Blöcke spannen. Manchmal ist eine einzubettende Nachricht beispielsweise zu groß, um in ein einziges visuelles Bild eingebettet zu werden. In diesem Fall kann die Nachricht in eine Anzahl von Teilstücken geteilt werden und jedes Teilstück kann in ein anderes visuelles Bild eingebettet werden. Diese verschiedenen visuellen Bilder können gelesen werden und die eingebetteten Nachrichten können ermittelt und miteinander verbunden werden, um die endgültige, gewünschte Nachricht zu bilden. Figur 14A illustriert schematisch eine Ausführungsform eines Bildes 1400 mit einer eingebetteten Nachricht, in der sowohl das visuelle Bild als auch die eingebettete Nachricht in drei benachbarte Teile 1401, 1402, 1403 geteilt sind. Unter Verwendung der drei Blöcke kann man beispielsweise die folgende Nachrichtendaten:

"http:\\www.glyph.com\\serial_blocks.html" in das gesamte Bild 1400 einbetten. Zu diesem Zweck kann man die Nachricht in drei Blöcke teilen und jeden in ein anderes Bild der Blöcke 1401, 1402, 1403 einbetten. So kann man die Daten "http:\\www.gl" in den ersten Block 1401, die Daten "yph.com\\seria" in den zweiten Block 1402 und die Daten "l_blocks.html" in den dritten Block 1403 einbetten. Als Ergebnis zeigt das erste Bild, d.h. der erste Teil (oder Block) 1401 des gesamten visuellen Bildes 1400 **"THREE"** (Figur 14B), das zweite Bild 1402 zeigt **"BLOCK"** (Figur 14C) und das dritte Bild 1403 zeigt **"IMAGE"** (Figur 14D); diese haben jeweils umkehrbare Pixel im Vordergrundbild, was die Existenz einer eingebetteten Nachricht anzeigt. Bei visueller Inspektion zeigen die drei Bilder in den Blöcken 1401, 1402, 1403 das gesamte Bild **"THREE BLOCK IMAGE"** (s. Figur 14E). In jedem Block kann ein Fahnenabschnitt vorhanden sein, um den Block mit den anderen Blöcken zu verbinden. Der Syncstrom im Block kann beispielsweise eine Fahne am Anfang beinhalten, die von einem Fehlerkorrekturstrom gefolgt wird. Der zweite und der dritte Block enthalten ebenfalls Fahnenab-

schnitte, um die Blöcke miteinander zu verbinden. Wenn sie in Reihe, d.h. nacheinander zusammengeschaltet sind, ist die gesamte Nachricht "http:\\www.glyph.com\\serial_blocks.html".

Figur 14F zeigt eine Ausführungsform, in der die Blöcke parallel eingebettete Nachrichten haben, d.h. jeder Block hat dieselbe eingebettete Nachricht, obwohl das visuelle Bild der einzelnen Blöcke unterschiedlich ist. So kann z.B. im Glyphblock 1410 das durchschnittliche Vordergrundbild (mit der eingebetteten Nachricht) in vier Blöcke unterteilt werden: erster Block 1414, zweiter Block 1415, dritter Block 1416 und vierter Block 1417.

Figuren 14G bis 14J zeigen die individuellen Blöcke der vier parallelen Blöcke aus Figur 14F. Figur 14K ist der gesamte resultierende Block, der aus den parallelen Blöcken aus Figur 14F besteht. In dieser Ausführungsform des parallelen Verfahrens ist die eingebettete Nachricht in jedem Block 1414, 1415, 1416, 1417 "http:\\www.glyph.com\\tiled_glyph_blocks". Es gibt also eine Informationsredundanz. Das Lesen eines jeden der Blöcke 1414, 1415, 1416, 1417 liefert die Nachricht "http:\\www.glyph.com\\tiled_glyph_blocks". Wenn ein Fehler in einem der Blöcke ist, z.B. einen Fehler der dadurch verursacht wird, daß ein Teil von Block 1410 bedeckt ist, wird durch die Redundanz eine Korrektur der defekten Information ermöglicht, um die akkurate Nachricht bereitzustellen. In jedem der Blöcke ist auch ein Fahnenabschnitt, der anzeigt, daß diese Blöcke parallele Blöcke sind.

Weiterhin können mehrere Glyphblöcke, von denen jeder ein separates Vordergrundbild und eine separate eingebettete Nachricht hat, zusammengefügt werden, um ein bedeutungsvolles, verbundenes Vordergrundbild mit entsprechenden eingebetteten

20.02.01

DE 199 83 484 T 1

Nachrichten zu bieten. Als weitere Illustration zeigt Figur 15 ein Beispiel von Codes einer HTML-Seite 1500, die vier durch Hyperlink verbundene URLs enthält:

- (1) <http://www.netshopper.com/>"
- (2) <http://www.netshopper.com/computers.html>"
- (3) <http://www.netshopper.com/monitors.html>"
- (4) <http://www.netshopper.com/printers.html>"

Dies wäre ein Beispiel für eine Website zum Verkauf von Computern und Zubehör über das Internet. In einer traditionellen Website, die mit dem Programm HTML aus Figur 15 codiert wurde, wird die durch Hyperlink verbundene Anzeige auf dem Monitor als Webseite 1502 wie in Figur 16 dargestellt, angezeigt. Eine Person, die die Seite mit einem Computer besucht, braucht nur mit einem Zeiger (z.B. einer Maus) beispielsweise auf den Hotspot "**Monitors**" klicken, um zu bewirken, daß der Browser mit der Webseite <http://www.netshopper.com/monitors.html> verbindet. Das gleiche HTML-Programm von Figur 15 kann codiert werden, um z.B. auf einem ausgedruckten Blatt Papier eine Webpage 1504 wie in Figur 17 dargestellt, darzustellen. In Figur 17 beinhaltet jeder der Glyphblöcke ein visuelles Bild mit eingebetteter Webadresse. Im Block 1506 hat das Bild "**NetShopper**" die eingebettete Nachricht <http://www.netshopper.com/>, in Block 1508 im Bild "**Computers**" ist die Nachricht <http://www.netshopper.com/computers.html> eingebettet, im Block 1510 "**Monitors**" ist die Nachricht <http://www.netshopper.com/monitors.html> eingebettet. Im Block 1512 "**Printers**" ist die Nachricht <http://www.netshopper.com/printers.html> eingebettet. Um mit einer Internetseite entsprechend einem der Hotspots aus Figur 17 zu verbinden, wenn einer der Blöcke 1506, 1507, 1508, 1510 mit einer Lesevorrichtung gemäß der vorliegenden Erfindung gelesen wird, bewirkt die Lesevorrichtung, daß ein Prozessor

(z.B. ein Computer) den Internetbrowser anstößt, mit der URL, entsprechend der Website zu verbinden. Es ist zu beachten, daß jeder der Blöcke (Hotspots) von Figur 17 ein deutlich erkennbares (und für menschliche Augen lesbares) Vordergrundbild mit bemerkbaren aber nicht störenden Punkten, die auf das Vorhandensein einer eingebetteten Nachricht (URL-Adresse) hindeuten, enthält. Wie groß die Blöcke sein müssen, hängt von der Qualität des Druckers und der Lesevorrichtung ab. Heutzutage erlauben es die vorhandenen Bildsensortechnologie und Drucker (z.B. 600 dpi) z.B. ein Bild wie in Figur 17 dargestellt in der Schriftart Time New Roman mit der Größe von 12 Punkt leicht zu lesen. Das im Handel erhältliche Offsite-Drucken ermöglicht viel feinere Drucke und somit kleinere Bilder, die die gleiche Information zu enthalten.

Der Prozeß des Druckens von eingebetteten URLs von HTML Seiten

Normalerweise sind die URLs einer HTML-Seite verloren, nachdem die Seite gedruckt wurde. Mit dem vorliegenden Verfahren können die URLs einer HTML-Seite immer noch vorhanden sein, nachdem die Seite auf Papier gedruckt wurde. Der Prozeß der Druckens eingebetteter URLs in einem lebenden Hotspot auf Papier von einer Standard-HTML-Seite kann ganz automatisch und transparent für die Benutzer funktionieren. Es gibt in diesem Prozeß 5 Schritte, die wie folgt sind. 1) Analysiere eine HTML-Seite, um alle Hotspots zu finden. 2) Finde die Position und die Abmessung von jedem Hotspot. 3) Konvertiere jeden Hotspot in ein echtes Vordergrundbild. 4) Codiere die URL-Adresse, die mit Hotspot assoziiert ist, in das relevante Vordergrundbild. 5) Drucke die Hotspots mit eingebetteter URL anstelle der regulären Hotspots auf Papier. Dieser Prozeß hat drei große Vorteile. Erstens muß die HTML-Seite nicht geändert werden. Zweitens muß das Layout der HTML-Seite nicht geändert

200001
DE 199 83 484 T1

werden. Drittens signalisieren die wahllos entstandenen Punkte auf dem Hotspot das Vorhandensein einer eingebetteten URL.

Integration des Codierens, Drucken des Bildes und Decodieren

In der Anwendung, die das vorliegende Verfahren des Codierens und Decodierens verwendet, wird eine Nachricht (die im allgemeinen nicht durch visuelle Inspektion decodiert werden kann) in ein visuelles Vordergrundbild auf eine Art und Weise eingebettet, die nicht störend ist, so daß das visuelle Bild nicht wesentlich verändert wird. Im allgemeinen kann das Bild auf die Oberfläche eines Substrats wie z.B. Papier gedruckt werden. Das visuelle Bild mit der eingebetteten Nachricht kann dann gelesen werden, z.B. durch Einscannen des Bildes mit einem Scanner in einen Prozessor, wie z.B. einen elektronischen Computer, z.B. einen Desktop-PC.

Figur 20A zeigt ein Beispiel eines Apparats zum Codieren einer Nachricht in ein visuelles Bild und Darstellen des visuellen Bildes mit der eingebetteten Nachricht. Die Codiervorrichtung 2000 beinhaltet einen Prozessor 2002, der einen Codieralgorithmus hat. Der Prozessor 2002 kann ein elektronischer Computer, ein Mikroprozessor o.ä. sein. Der Prozessor 2002 kann sowohl Codes zum Codieren des visuellen Bildes als auch der eingebetteten Nachricht gemäß dem Verfahren der vorliegenden Erfindung aufweisen. Ein Benutzerinterface 2004 ist mit dem Prozessor 2002 verbunden, um Parameter, Daten, Programme, Programmmodifikationen und Algorithmen, Nachrichten, Bildbearbeitungen usw. in den Prozessor 2002 einzugeben. Beispiele für das Benutzerinterface 2004 sind eine Tastatur, eine Zeigevorrichtung (z.B. eine Maus), ein elektronischer Bleistift, sprachgestützte Eingabevorrichtungen usw. Der Prozessor 2002 selbst kann einen Speicher zum Speichern der Algorithmen, Pro-

gramme, Daten usw. haben; es kann aber auch ein Speicher 2006 mit dem Prozessor für ein solches Speichern verbunden sein. Das visuelle Bild mit der(n) eingebetteten Nachricht(en) kann sowohl zur visuellen Betrachtung als auch zum Importieren in eine Lesevorrichtung durch eine Darstellvorrichtung 2008 dargestellt werden. Beispiele für die Darstellvorrichtung 2008 sind ein Elektronenstrahlmonitor, eine LCD-Anzeige, ein Drucker u.ä. Im Falle eines Druckers kann man einen Ausdruck (Hardcopy) 2010 mit dem visuellen Bild, das die eingebettete Nachricht einschließt durch Drucken auf ein Medium, z.B. Papier, erhalten. Optional kann ein Benutzerinterface 2004 mit der Darstellvorrichtung 2008 verbunden werden, um die Darstellvorrichtung 2008 zu steuern. Elektronische Verbindungen können durch Kabel, Drähte usw. hergestellt werden. Schnurlose Verbindungen zwischen den Geräten können ebenfalls zur Signalübertragung, z.B. über elektromagnetische Wellen, wie Mikrowellen oder Infrarotsignalübertragung realisiert werden. Ein Beispiel für eine solche Verwendung der Vorrichtung ist es auf ein Papier ein visuelles Bild zu drucken, z.B. das Bild "Webstar", in das die URL `http:\\www.Webstar.com\\` eingebettet ist.

Figur 20B zeigt eine Ausführungsform einer Lesevorrichtung zum Lesen der eingebetteten Nachricht von einem visuellen Bild, wie z.B. einem Bild, das auf der Darstellvorrichtung 2008 dargestellt ist. Die Lesevorrichtung 2014 beinhaltet einen Imager (Bildsensor) 2016, der das visuelle Bild, das auf der Darstellvorrichtung dargestellt ist, registriert. Wenn die Darstellvorrichtung ein Drucker ist, der eine Hardcopy herstellt, dann kann der Imager 2016 beispielsweise jede Vorrichtung sein, z.B. ein Scanner, CMOS-Sensor oder eine ladungsgekoppelte Schaltung CCD, die die Pixel (d.h. die Bildelemente) in elektrische Signale konvertiert. Ein Prozessor 2018 konver-

tiert die elektrischen Signale in Pixelwerte und in die digitalen Werte 0 und 1, um die Datenstruktur einer Bitmap herzustellen. Der Prozessor 2018 verwendet die Algorithmen zum Decodieren der Bitmapdaten in den Syncstrom, Datenstrom und schließlich zum Decodieren der eingebetteten Nachricht. Der Prozessor 2018 kann einen eigenen Speicher haben oder kann mit einem Speicher 2020 zum Speichern der Daten, Parameter, Programme, Algorithmen usw. verbunden sein. Der Prozessor 2018 kann ein elektronischer Computer, Mikroprozessor o.ä. sein. Optional kann ein Benutzerinterface 2022 verwendet werden, um Daten, Parameter, Programme, Bearbeitungen usw. in den Prozessor 2018 einzugeben. Beispiele für das Benutzerinterface 2004 sind eine Tastatur, eine Zeigevorrichtung (z.B. eine Maus), ein elektronischer Bleistift, sprachgestützte Eingabevorrichtungen usw. Ebenfalls optional kann das Benutzerinterface 2022 mit dem Imager 2016 verbunden sein, um den Imager 2016 zu steuern. Der Imager 2016 kann weiterhin eine Vorrichtung sein, die das Bitmapbild erfaßt mit einem Videoaufnahmegerät, wie einer Videokamera, CCD o.ä. Dies ist insbesondere anwendbar, um das Bild von einer lichtabstrahlenden Anzeigevorrichtung, wie einem Bildschirm zu erfassen. Imagers wie Scanner, Kameras usw. sind bekannt. Desktopscanner und Handscanner sind beispielsweise im Handel erhältlich und ihre Technologie ist bekannt.

Figur 20C stellt eine Ausführungsform der vorliegenden Erfindung zum Decodieren eines gedruckten Bildes mit einer eingebetteten URL und Aktivieren eines Computers, um einen Internetbrowser so zu lenken, so daß er auf die Website gemäß der URL zugreift, dar. In dieser Ausführungsform eines Internetzugriffsystems 2030 kann eine Fernsteuerungslesevorrichtung 2032 die eingebettete URL von einer ausgedruckten Seite 2034 lesen und diese Information einem entfernten TV/Computer 2038 kommu-

nizieren, der über einen Server mit dem Internet 2040 verbunden ist. Die Fernsteuerungslesevorrichtung 2032 hat beispielsweise eine LCD-Anzeige 2033, die die decodierte Nachricht darstellt. Die vorliegende Fernbedienung hat einen Bildsensor (oder Imager) 2016 und kann als Lesevorrichtung 2014 von Figur 20B funktionieren. Weiterhin kann der Computer 2038 ein Prozessor in einem Fernseher mit Internetzugang sein. Die Fernsteuerungslesevorrichtung 2032 kann eine Fernsteuerung für einen Fernseher sein, mit zusätzlichen Merkmalen zum Lesen von codierten Nachrichten in ein Bild wie hier beschrieben. Typischerweise hat die Fernsteuerungslesevorrichtung eine Batterie zur Energieversorgung und einen Sender zum Senden von Signalen aus elektromagnetischen Wellen zum Computer. Zusätzlich kann eine Zeigevorrichtung wie eine Maus in der Fernsteuerungslesevorrichtung 2032 vorgesehen sein, um eine Steuerung des Computers zu gewährleisten, um die normale Computersoftware und Computerprogramme laufen zu lassen. Die Technologie für traditionelle Fernsteuerungen, Internetverbindung, Internetbrowser und Zeigevorrichtungen sind im Stand der Technik bekannt und werden hier nicht beschrieben.

Figur 20D illustriert schematisch die Struktur einer Fernsteuerungslesevorrichtung 2032. Kurz gesagt hat die Fernsteuerungslesevorrichtung 2032 einen Mikroprozessor 2040, der die Kommunikation und die Datenverarbeitung steuert. Ein Fernsteuerungstastaturfeld gibt die Information in den Mikroprozessor 2040 ein, z.B. wird das Ein- oder Ausschalten einer Lichtquelle 2042 initiiert, um das Bild 2034, das auf eine Seite 2036 gedruckt ist, zu beleuchten. Ein Lichtsensor (z.B. eine CCD-Kamera, eine CMOS-Kamera, eine Photodetektionsanordnung oder ein Scanner) und ein A/D-Konverter (Block 2044) produzieren digitale, elektrische Signale, die von der Lichtintensität (und optional der Farbe, im Falle einer Farbkamera)

der Lichteinstrahlung auf dem Lichtsensor abhängen. Der Prozessor 2040 kann einen Speicher 2046 haben, der mit ihm zum Speichern von Information, Daten, Programmen usw. verbunden ist. Ein Decoder 2048 stellt das Decodieren des Signals, das der Mikroprozessor 2040 vom A/D-Konverter bekommt, um die URL zu identifizieren, bereit. Der Decoder 2048 enthält die Algorithmenprogramme zum Decodieren des visuellen Bildes, um die im visuellen Bild eingebettete Nachricht herzuleiten. Die URL-Adresse wird über den Kommunikationschip 2050 an den Computer 2038 kommuniziert. Der Kommunikationschip 2050 und der Mikroprozessor 2040 kommunizieren mit einem nichtflüchtigen Speicher in der Fernsteuerungslesevorrichtung 2032. Der nichtflüchtige Speicher 2032 stellt eine Datenspeicherkapazität zum Lesen einer eingebetteten URL von einem Bild auf einer gedruckten Seite, die weit weg vom Computer 2038 ist, bereit. Die Fernsteuerungslesevorrichtung 2032 kann später in die Nähe des Computers gebracht werden, um den Zugang des Internetbrowsers zur URL-Adresse zu aktivieren. Wenn also z.B. die Fernsteuerungslesevorrichtung 2032 ein visuelles Bild "Webstar" mit der eingebetteten URL von Figur 8A liest, dann überträgt sie die Daten zum Computer 2038, um den Internetbrowser zum Zugriff auf die Website mit der URL "<http://www.Webstar.com>" zu aktivieren.

Eine erfindungsgemäße Vorrichtung mit Zugriff auf das Internet kann ein Bild decodieren, das eine eingebettete Internetadresse aufweist, um mit dem Internet zu verbinden. In der Verwendung hier schließt der Begriff "Internet" das World Wide Web ein, aber beinhaltet auch Netzwerke zu Websites, die nur bestimmten autorisierten Personen zugänglich sind. Unabhängig vom Ausmaß der Zugänglichkeit ist die vorliegende Erfindung anwendbar für autorisierte Benutzer, um auf solche Sites zuzugreifen, indem die Website-Adresse von einem Bild mit einer

der Lichteinstrahlung auf dem Lichtsensor abhängen. Der Prozessor 2040 kann einen Speicher 2046 haben, der mit ihm zum Speichern von Information, Daten, Programmen usw. verbunden ist. Ein Decoder 2048 stellt das Decodieren des Signals, das der Mikroprozessor 2040 vom A/D-Konverter bekommt, um die URL zu identifizieren, bereit. Der Decoder 2048 enthält die Algorithmenprogramme zum Decodieren des visuellen Bildes, um die im visuellen Bild eingebettete Nachricht herzuleiten. Die URL-Adresse wird über den Kommunikationschip 2050 an den Computer 2038 kommuniziert. Der Kommunikationschip 2050 und der Mikroprozessor 2040 kommunizieren mit einem nichtflüchtigen Speicher in der Fernsteuerungslesevorrichtung 2032. Der nichtflüchtige Speicher 2032 stellt eine Datenspeicherkapazität zum Lesen einer eingebetteten URL von einem Bild auf einer gedruckten Seite, die weit weg vom Computer 2038 ist, bereit. Die Fernsteuerungslesevorrichtung 2032 kann später in die Nähe des Computers gebracht werden, um den Zugang des Internetbrowsers zur URL-Adresse zu aktivieren. Wenn also z.B. die Fernsteuerungslesevorrichtung 2032 ein visuelles Bild "Webstar" mit der eingebetteten URL von Figur 8A liest, dann überträgt sie die Daten zum Computer 2038, um den Internetbrowser zum Zugriff auf die Website mit der URL "<http://www.Webstar.com>" zu aktivieren.

Eine erfindungsgemäße Vorrichtung mit Zugriff auf das Internet kann ein Bild decodieren, das eine eingebettete Internetadresse aufweist, um mit dem Internet zu verbinden. In der Verwendung hier schließt der Begriff "Internet" das World Wide Web ein, aber beinhaltet auch Netzwerke zu Websites, die nur bestimmten autorisierten Personen zugänglich sind. Unabhängig vom Ausmaß der Zugänglichkeit ist die vorliegende Erfindung anwendbar für autorisierte Benutzer, um auf solche Sites zuzugreifen, indem die Website-Adresse von einem Bild mit einer

eingebetteten Nachricht gelesen wird. Der Mikroprozessor 2040 in der Fernsteuerungslesevorrichtung 2032 kann so eingestellt werden, daß die decodierte Website-Adresse sofort und automatisch vom Computer 2038 aufgerufen wird, um über das Internet mit der entsprechenden Seite zu verbinden. Alternativ kann die Website gespeichert werden, wenn sie von der Fernsteuerungslesevorrichtung 2032 gelesen wird, und kann später verwendet werden, um die entsprechende Adresse aufzurufen. Die Arbeitsweise der Fernsteuerungslesevorrichtung 2032 ist analog zu einer Person, die eine Maus verwendet, um auf dem Bildschirm auf einen Hotspot zu klicken, z.B. auf eine durch Hyperlink verbundene Website, um die Verbindung zu der durch Hyperlink verbundenen Website herzustellen. Der Unterschied ist, daß der Benutzer anstatt eine Maus (oder eine andere Zeigevorrichtung), um auf eine durch Hyperlink verbundenen Website auf dem Bildschirm zu klicken, im vorliegenden Websitedecodieren eine Fernsteuerungslesevorrichtung 2032 verwendet, um den Hotspot in dem Bild mit der eingebetteten Website-Adresse zu lesen oder zu scannen.

Die Codiervorrichtung 2000 und die Lesevorrichtung 2014 und ähnliche Vorrichtungen können zum Codieren und Decodieren gemäß dem Verfahren, wie oben beschrieben, verwendet werden. Figur 21A zeigt ein Ablaufdiagramm, das kurz beschreibt, wie eine URL-Adresse (die Nachricht) in ein Glyphbild eingebettet und dargestellt wird. Zunächst wird die URL einer HTML-Seite als die einzubettende Nachricht ausgewählt (Block 2012). Um zum Beispiel einen entfernten Link zur Website "Webstar" einzubetten, kann es gewünscht sein, die Nachricht "http:\\www.webstar.com\\" einzubetten. Es wird auch der Hotspot der HTML-Seite als Vordergrundstring ausgewählt (Block 2004). Das visuelle Bild, in das die Nachricht eingebettet werden kann, kann beispielsweise das Bild "Webstar" sein. Als näch-

stes wird die URL-Adresse codiert, um in den Vordergrundstring (das visuelle Bild) eingebettet zu werden, um ein Glyphbild (Block 2006) herzustellen. Das Glyphbild wird z.B. durch Drucken auf Papier oder Anzeigen auf einem Computerbildschirm dargestellt (Block 2008), anstelle eines regulären Hotspots. Der Drucker kann beispielsweise verwendet werden, um einen Ausdruck der URL-Adresse, die in dem Bild auf dem Papier eingebettet ist, zu drucken.

Der Prozeß, wie die eingebettete URL-Adresse aus dem Glyphbild gelesen wird, wird kurz in dem Ablaufdiagramm von Figur 21B dargestellt. Ein dargestelltes Glyphbild, z.B. ein auf einem Papier ausgedrucktes Glyphbild, wird von einer zweidimensionalen Lesevorrichtung gelesen, z.B. der Fernsteuerungslesevorrichtung 2032 der vorliegenden Erfindung (Block 2112). Das Glyphbild wird decodiert, um die eingebetteten Nachrichtendaten wiederherzustellen (Block 2114). Optional kann die Information sofort an einen Computer gesendet werden, um auf die eingebettete Nachricht, die decodiert wurde, zu antworten (Block 2116). Die Lesevorrichtung kann beispielsweise einen Internetbrowser aktivieren, um die HTML-Seite herunterzuladen oder auf die gewünschte Website gemäß der decodierten eingebetteten Nachricht, d.h. die URL der Website, zu schalten.

Die vorliegende Erfindung wurde in der obigen Beschreibung beschrieben. Die bevorzugten Ausführungsformen dienen nur illustrativen Zwecken und sollen nicht den Umfang der Erfindung einschränken. Die Beschreibung sollte so verstanden werden, daß für Fachmann Modifikationen und Änderungen der Erfindung offensichtlich sind, ohne daß diese vom Umfang der Erfindung abweichen. Die Algorithmen oder Programme können beispielsweise in der Fernsteuerungslesevorrichtung vorhanden sein, in dem TV/Computer, einem Medium zum Speichern von Programmen,

20.02.01

DE 199 83 484 T 1

z.B. einer Floppydisk, einer Festplatte, einem Band, einer CD
usw.

20.02.01

199 83 484.9

58

DE 199 83 484 T1

Kenneth H.P. Chang

8983.PT-DE

PM/TE

Zusammenfassung

Verfahren zum Decodieren einer Nachricht, die in einem Pixelmuster eingebettet ist. Das Verfahren umfaßt die Schritte des Bestimmens der Pixelwerte für die Pixel von dem Pixelmuster; Bestimmen der binären Werte aus den Pixelwerten für die Pixel des Pixelmusters; und Bestimmen der eingebetteten Nachricht aus den binären Werten. Die Pixel haben einen Pixelwertebereich zwischen einem Maximum und einem Minimum. Die Pixel sind in Zellen unterteilt, die jeweils eine Glyphzelle und Hintergrundpixel haben. Der binären Wert eines Glyphpixels wird durch den Kontrast, den das Glyphpixel mit den Hintergrundpixeln hat, bestimmt. Dieses Verfahren kann verwendet werden, um eine eingebettete Website-Adresse von einem Bild mit einem Vordergrundbild und der eingebetteten Website-Adresse zu decodieren.

(Figur 1)

20.02.01

sq

DE 199 83 4 84 T 1

Kenneth H.P. Chang

8983.PT-DE

PM/TE

Patentansprüche

1. Vorrichtung zum Laden einer Website-Adresse in einen Internetbrowser, um mit der Website zu verbinden, die folgendes aufweist:
 - (a) eine Lesevorrichtung zum Erhalten von Licht von einem Pixelmuster und Bestimmen des Pixelwerts von jedem Pixel, wobei das Pixelmuster ein visuelles Vordergrundbild darstellt, das Informationen an einen Betrachter weitergibt; und
 - (b) einen Prozessor, der ein Programm mit Codiermitteln, die vom Prozessor gelesen werden können, um eine eingebettete Website-Adresse aus dem Pixelmuster wiederherzustellen, aufweist, wobei das Programm folgendes aufweist:
 - (i) Codiermittel zum Bestimmen der eingebetteten Website-Adresse durch Bestimmen von binären Werten unter Verwendung des Helligkeitskontrasts, um Pixelwerte von ausgewählten Pixeln mit den Pixelwerten von Nachbarpixeln im Pixelmuster zu vergleichen; und
 - (ii) Codiermittel zum Laden der Website-Adresse in den Internetbrowser zum Verbinden mit der Website gemäß der Website-Adresse.
2. Vorrichtung nach Anspruch 1, wobei die Lesevorrichtung eine Lichtquelle zum Beleuchten des Pixelmusters aufweist, um ein Lichtmuster von verschiedenen Pixelwerten bereitzustellen.

len, und die Lesevorrichtung einen Bildsensor aufweist, um das Licht aufzunehmen.

3. Vorrichtung nach einem der Ansprüche 1 bis 2, wobei das Codiermittel zum Bestimmen der eingebetteten Website-Adresse das Pixelmuster in Zellen teilt, die jeweils eine Pixelmatrix enthalten, um die binären Werte zu bestimmen.
4. Vorrichtung nach Anspruch 3, wobei das visuelle Bild einen Text oder ein Objekt zeigt und wobei das Codiermittel zum Bestimmen der eingebetteten Website-Adresse einen binären Wert eines Pixels in einer Zelle durch Vergleichen seines Pixelwerts mit einem Wert, der von den Pixelwerten der Pixel in seiner Nachbarschaft in der Zelle hergeleitet wurde, bestimmt.
5. Vorrichtung nach einem der Ansprüche 3 bis 4, wobei das Codiermittel zum Bestimmen der eingebetteten Website-Adresse weiterhin Daten von Pixeln in einen Bereich des Lichtmusters mit Daten von Pixeln in einem anderen Bereich des Lichtmusters auf Symmetrie vergleicht, um die eingebettete Website-Adresse zu bestimmen.
6. Vorrichtung nach einem der Ansprüche 3 bis 5, wobei das Codiermittel zum Bestimmen der eingebetteten Website-Adresse die Pixel in jeder Zelle in ein oder mehrere Glyphpixel und Hintergrundpixel separiert und den binären Wert bestimmt durch Bestimmen der Beziehung zwischen den Daten eines Glyphpixels in einer Zelle und den Daten eines Glyphpixels in einer anderen Zelle.
7. Vorrichtung nach einem der Ansprüche 1 bis 6, die weiterhin einen Computer zum Betreiben des Internetbrowsers und eine

Übertragungsvorrichtung zum Übertragen von elektromagnetischen Signalen, die die Website-Adresse darstellen, zum Computer, um den Internetbrowser so zu lenken, daß er mit der Internetseite verbindet, aufweist.

8. Verfahren zum Laden einer URL-Adresse in einen Internetbrowser, das folgendes aufweist:
 - (a) Erhalten von Licht von einer Anzeige, die ein Pixelmuster zeigt, und Bestimmen des Pixelwerts eines jeden Pixels, wobei das Pixelmuster ein visuelles Vordergrundbild bildet, das Informationen an einen Betrachter weitergibt;
 - (b) Wiederherstellen einer eingebetteten URL-Adresse durch Bestimmen binärer Werte unter Verwendung des Helligkeitskontrasts, um Pixelwerte von ausgewählten Pixeln mit den Pixelwerten der Nachbarpixel in dem Pixelmuster zu vergleichen;
 - (c) Laden der URL-Adresse in den Internetbrowser zum Verbinden mit der Website gemäß der URL-Adresse.
9. Verfahren nach Anspruch 8, das weiterhin das Anordnen der Pixel in dem Pixelmuster in Zellen, von denen jede eine Pixelmatrix hat, wobei jede Zelle ein oder mehrere Glyphpixel und benachbarte Hintergrundpixel hat; und für jedes Glyphpixel das Bestimmen des binären Werts des Glyphpixels auf der Grundlage des Vergleichens der Helligkeit des Glyphpixels mit einem Wert, der aus der Helligkeit der Hintergrundpixel in der Zelle abgeleitet wurde, aufweist, und wobei das visuelle Bild einen Text oder ein Objekt zeigt.
10. Verfahren zum Decodieren einer Nachricht, die in einem Pixelmuster eingebettet ist, wobei das Verfahren folgendes aufweist:

- (a) Bestimmen des Pixelwerts basierend auf der Helligkeit für Pixel aus dem Pixelmuster, wobei das Pixelmuster ein visuelles Vordergrundbild bildet, das Informationen an einen Betrachter weitergibt;
- (b) Bestimmen der binären Werte aus den Pixelwerten für Pixel aus dem Pixelmuster durch Vergleichen der Helligkeit ausgewählter Pixel mit der Helligkeit der benachbarten Pixel;
- (c) Bestimmen der eingebetteten Nachricht aus den binären Werten.

- Leerseite -

85

1/22

100

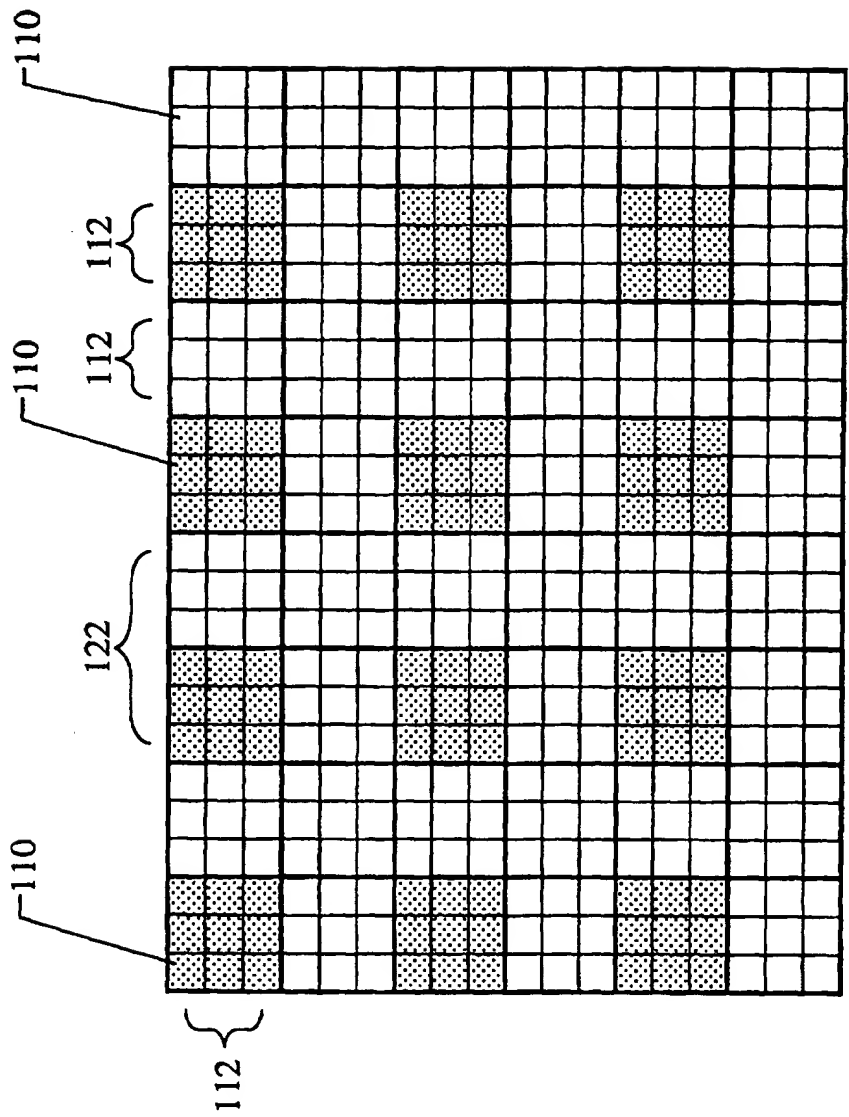


FIG. 1

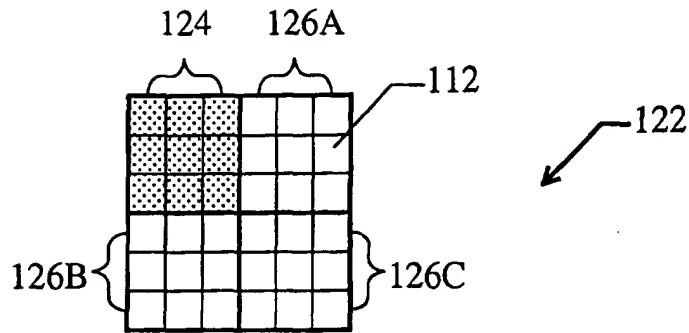


FIG. 2

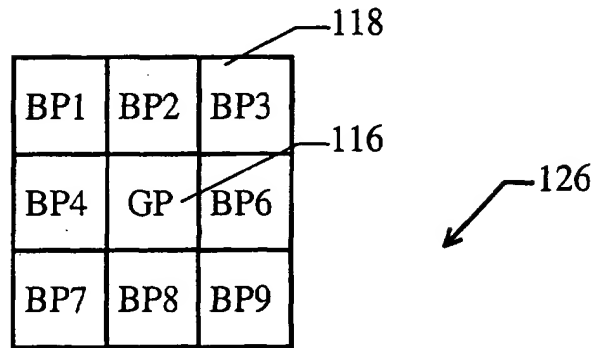


FIG. 3A

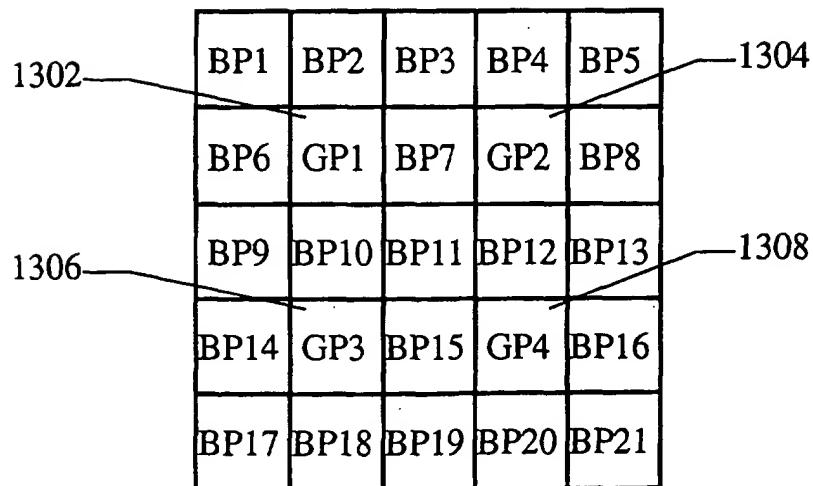


FIG. 3B

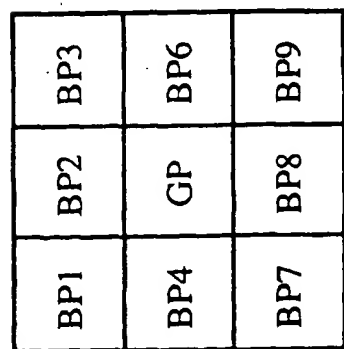


FIG. 4A

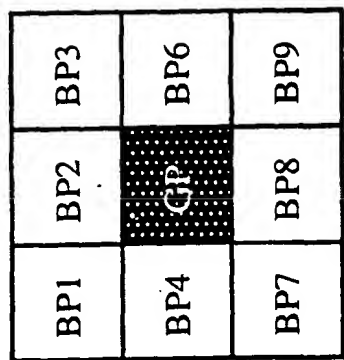


FIG. 4B

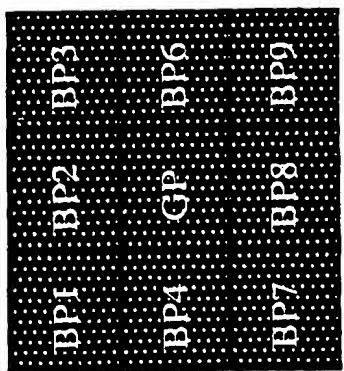


FIG. 4C

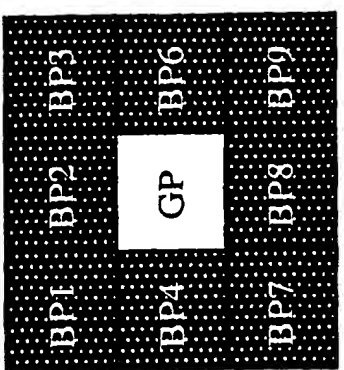


FIG. 4D

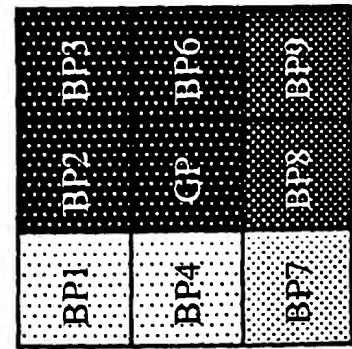


FIG. 4E

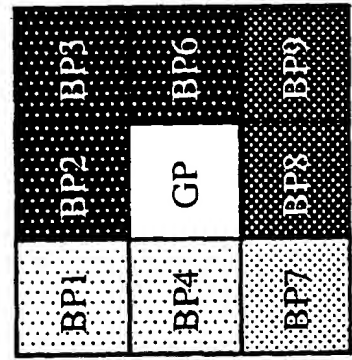


FIG. 4F

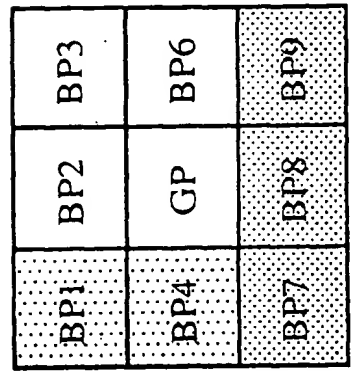


FIG. 4G

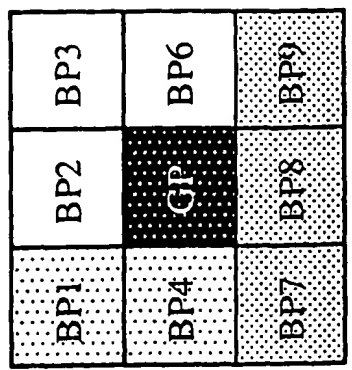


FIG. 4H

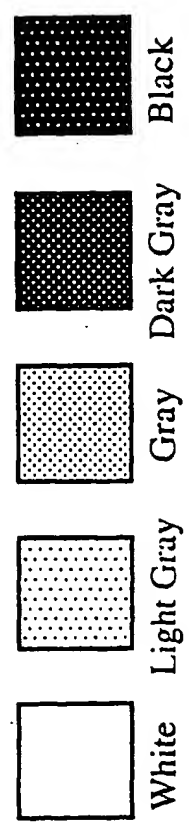


FIG. 4I

66 20.02.01

DE 199 83 484 T1

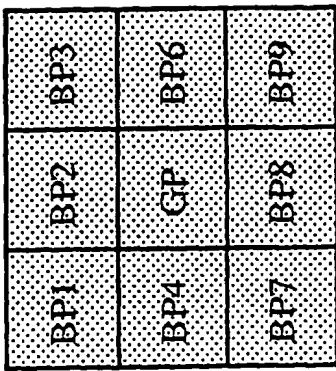


FIG. 5D

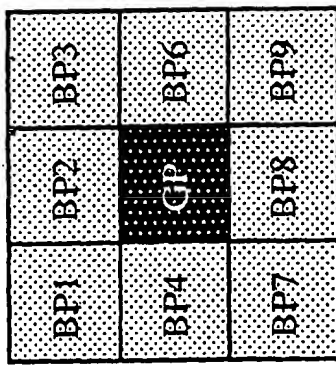


FIG. 5C

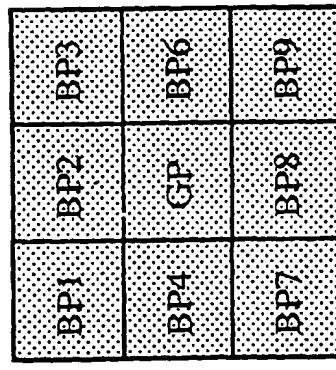


FIG. 5B

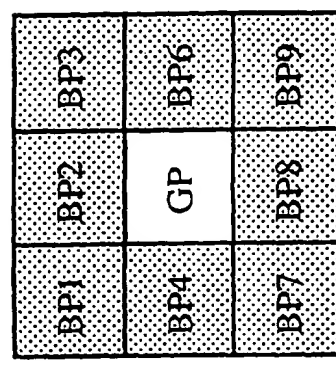


FIG. 5A

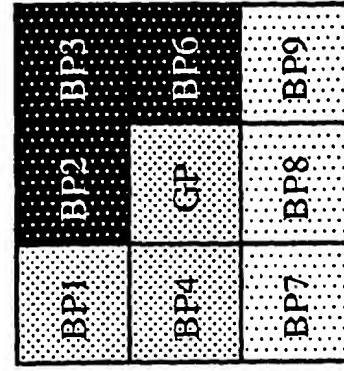


FIG. 5H

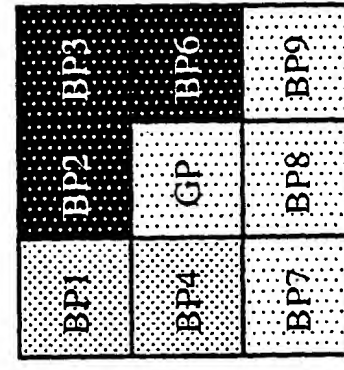


FIG. 5G

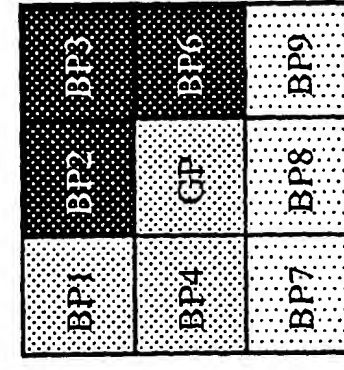


FIG. 5F

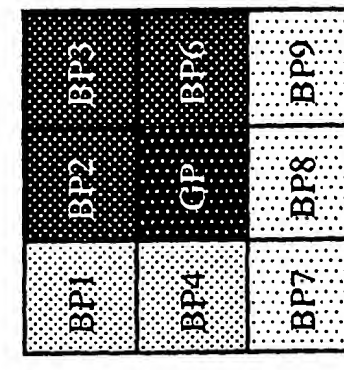


FIG. 5E

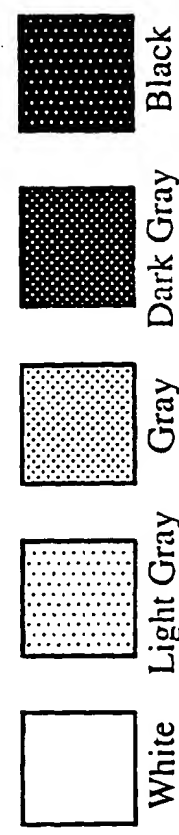


FIG. 5I

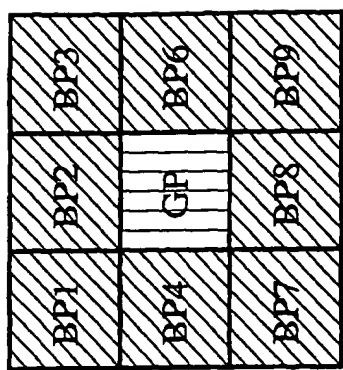


FIG. 6D

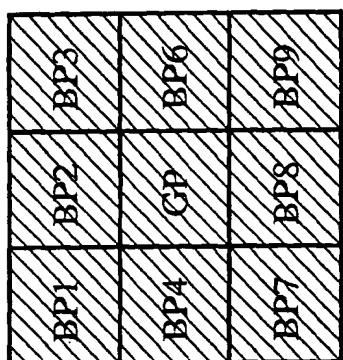


FIG. 6C

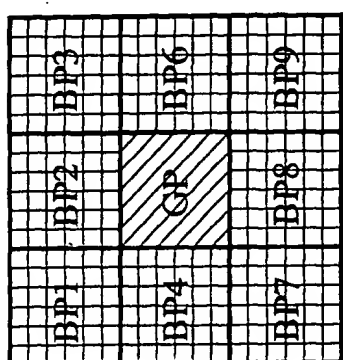


FIG. 6B

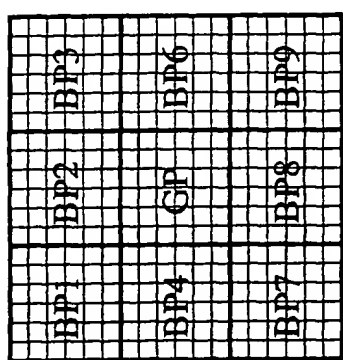


FIG. 6A

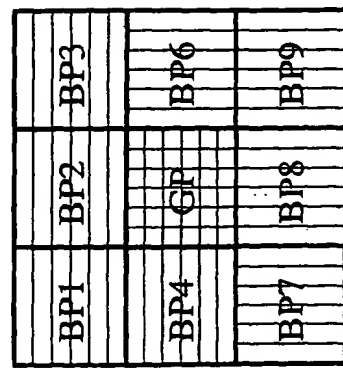


FIG. 6H

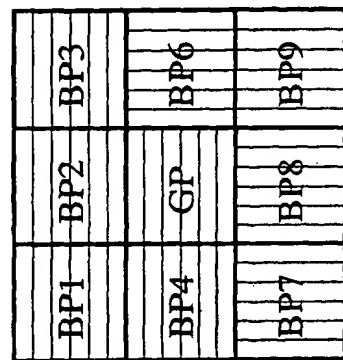


FIG. 6G

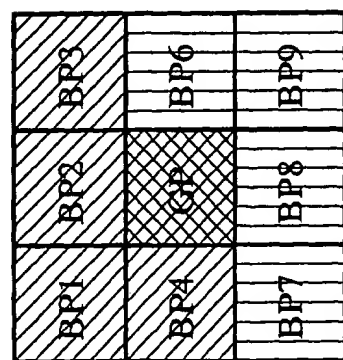


FIG. 6F

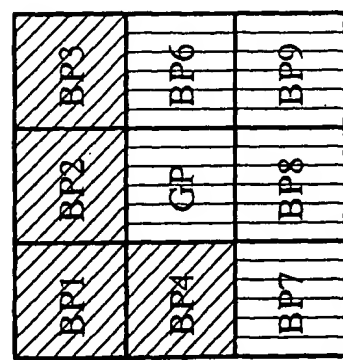


FIG. 6E

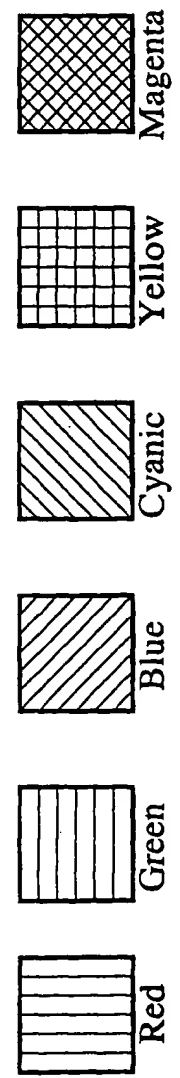


FIG. 6I

Webstar

FIG. 7

68

20.02.01
DE 199 83 4 84 T 1

6/22

69 20.02.01
DE 199 83 484 T1

7/22

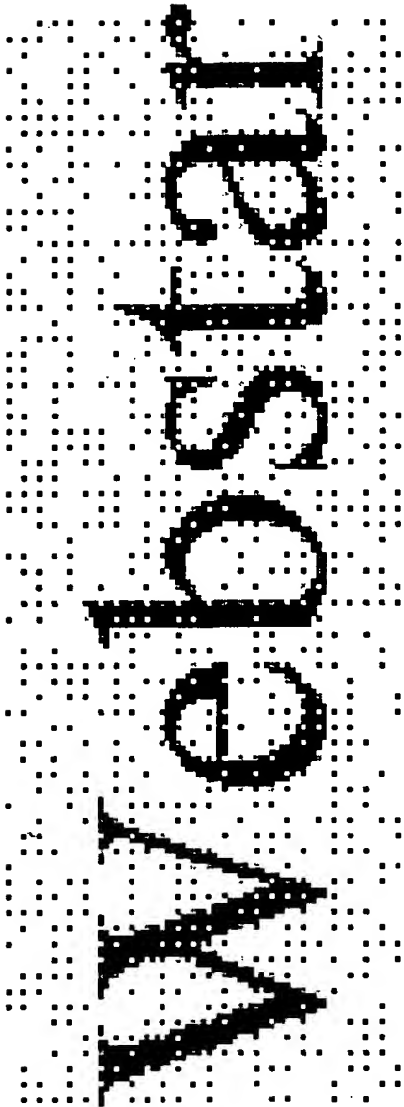


FIG. 8A

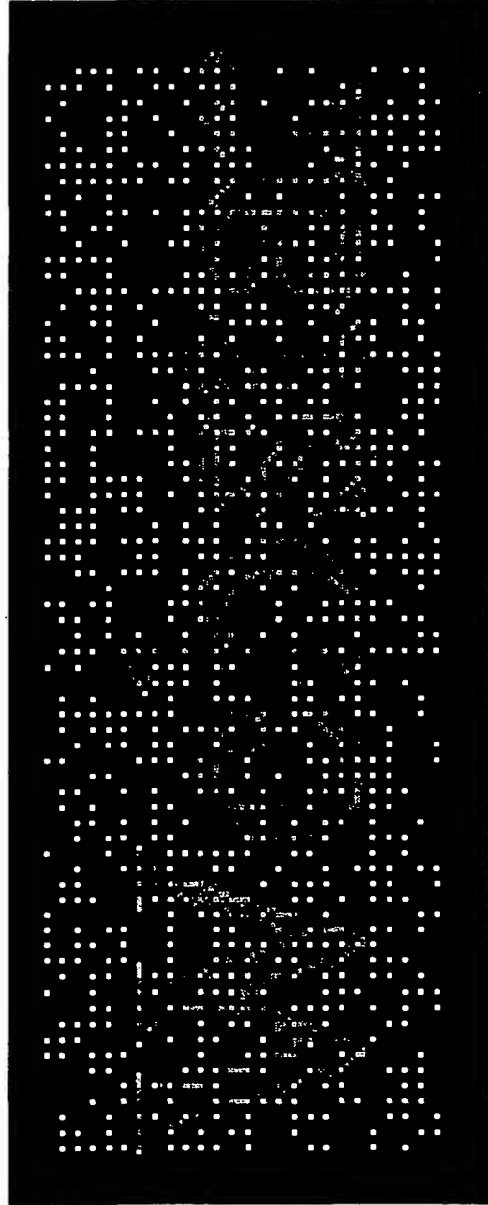
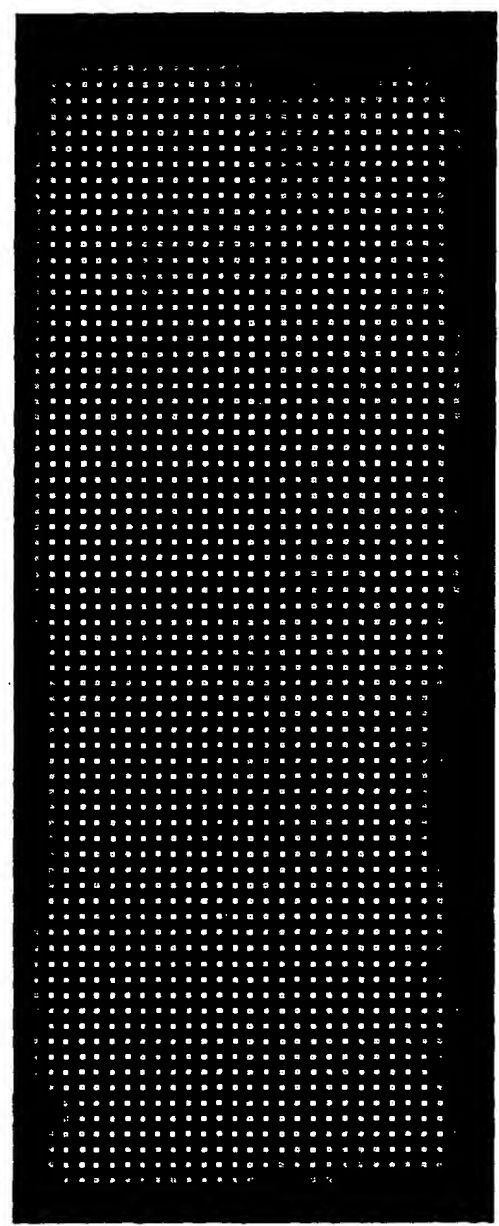
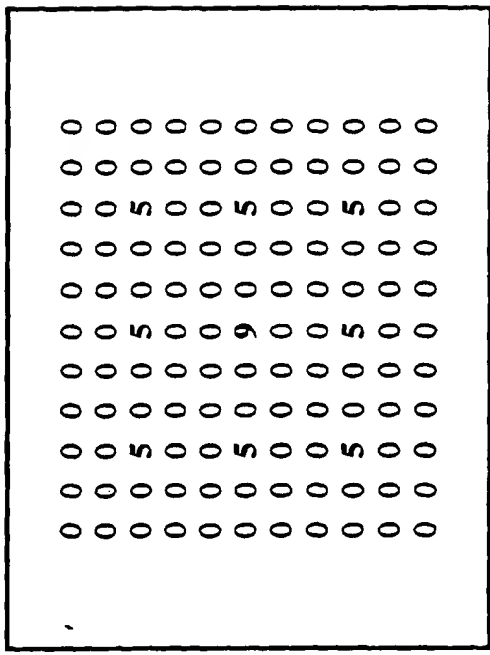


FIG. 8B

70 20.02.01
DE 199 83 484T1

8/22



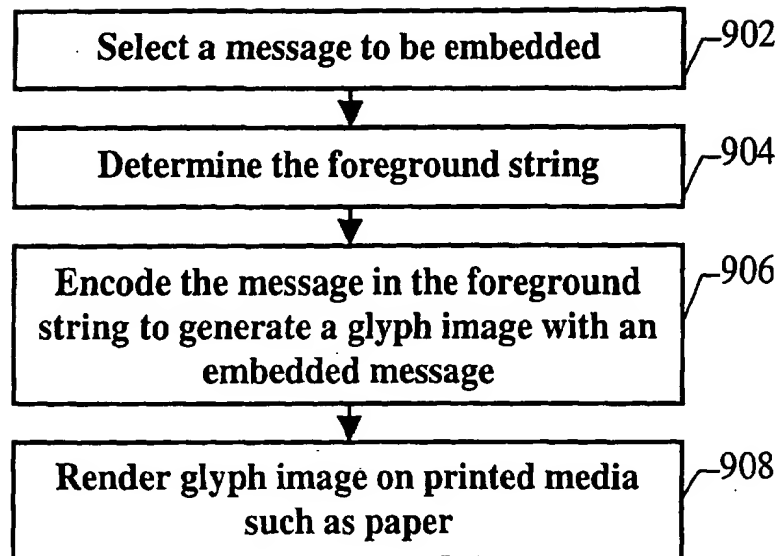


FIG. 9A

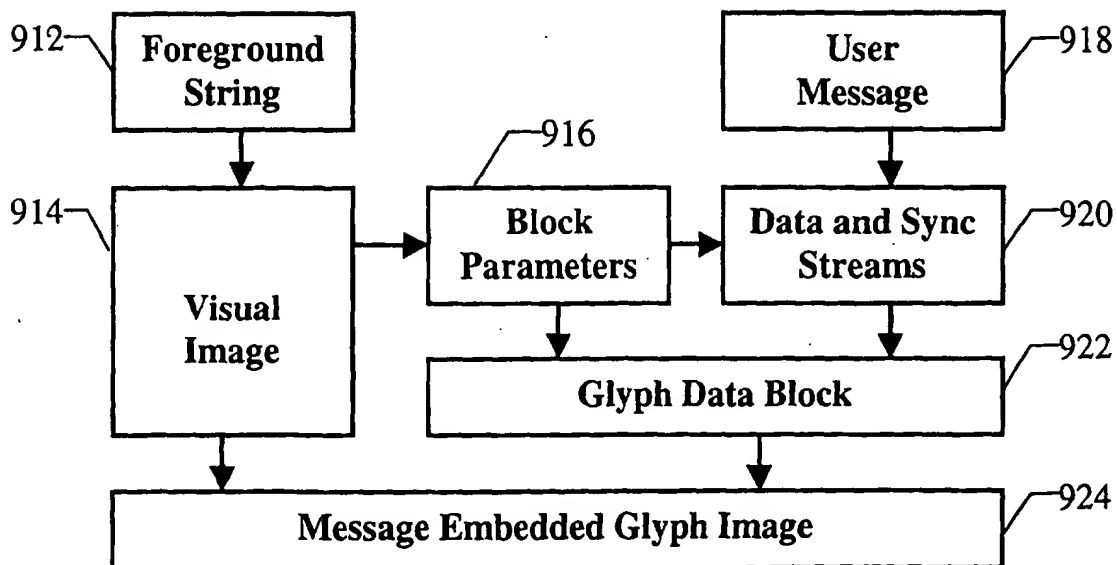


FIG. 9B

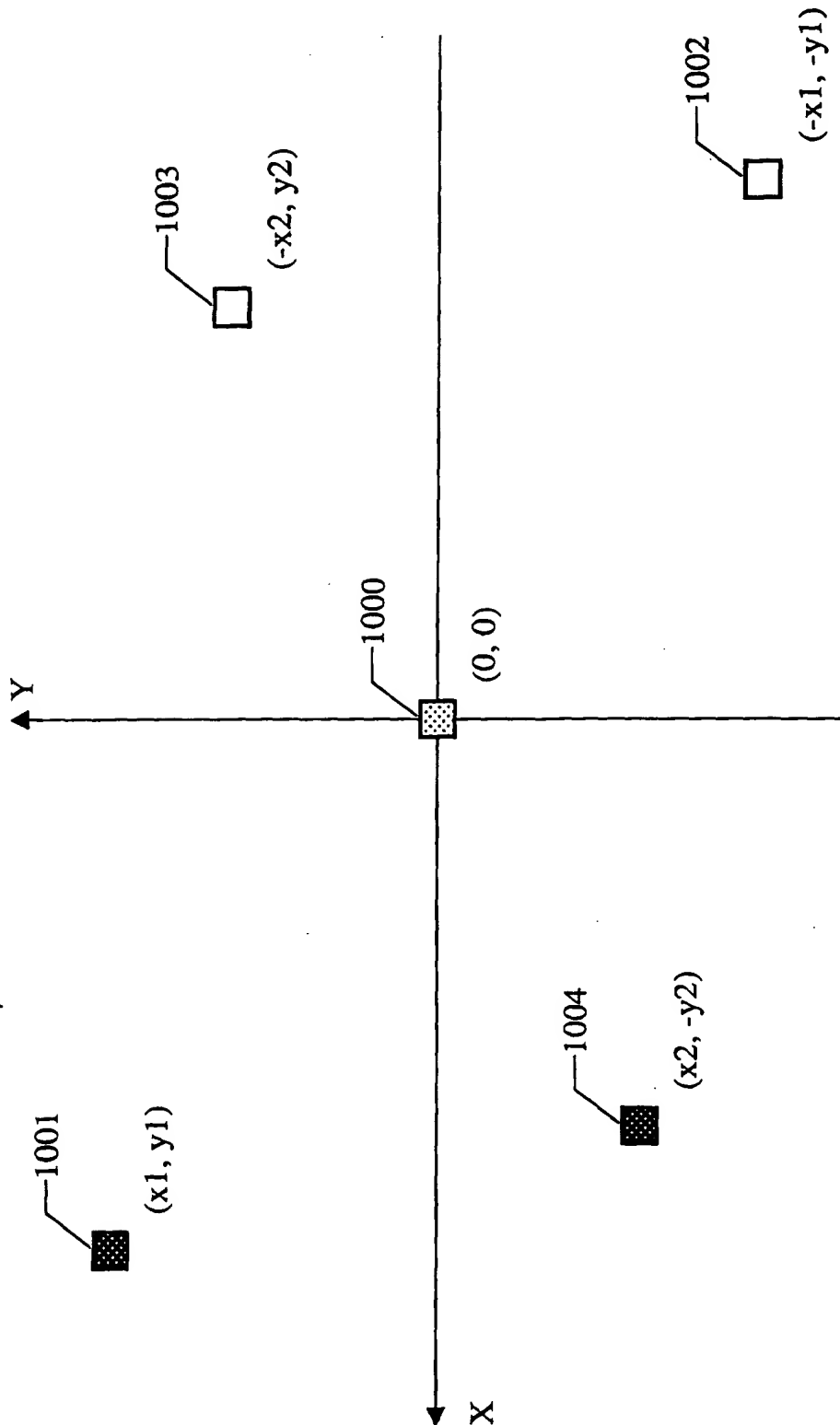


FIG. 10

d1	d2	d3	d4	d5	d6	d7	d8	d9	d10	d11	d12	d13	d14	d15	d16	d17	d17	d19	d20	d21	d22	d23	d24	d25
d26	d27	d28	d29	d30	d31	d32	d33	d34	d35	d36	d37	d38	d39	d40	d41	d42	d43	d44	d45	d46	d47	d48	d49	d50
d51	d52	d53	d54	d55	d56	d57	d58	d59	d60	d61	d62	d63	d64	d65	d66	d67	d68	d69	d70	d71	d72	d73	d74	d75
d76	d77	d78	d79	d80	d81	d82	d83	d84	d85	d86	d87	x	d87	d86	d85	d84	d83	d82	d81	d80	d79	d78	d77	d76
d75'	d74'	d73'	d72'	d71'	d70'	d69'	d68'	d67'	d66'	d65'	d64'	d63'	d62'	d61'	d60'	d59'	d58'	d57'	d56'	d55'	d54'	d53'	d52'	d51'
d50'	d49'	d48'	d47'	d46'	d45'	d44'	d43'	d42'	d41'	d40'	d39'	d38'	d37'	d36'	d35'	d34'	d33'	d32'	d31'	d30'	d29'	d28'	d27'	d26'
d25'	d24'	d23'	d22'	d21'	d20'	d19'	d18'	d17'	d16'	d15'	d14'	d13'	d12'	d11'	d10'	d9'	d8'	d7'	d6'	d5'	d4'	d3'	d2'	d1'

1104

1100

1102

1106

FIG. 11

1201	s1	d1	s2	d4	s3	d7	s4	d10	s5	d13	s6	d16	s7	d19	s8	d22	s9	d25	s10	d28	s11	d31	s12	d34	s13	d37
	d2	d3	d5	d6	d8	d9	d11	d12	d14	d15	d17	d18	d20	d21	d23	d24	d26	d27	d29	d30	d32	d33	d35	d36	d38	d39
1201	s14	d40	s15	d43	s16	d46	s17	d49	s18	d52	s19	d55	s20	d58	s21	d61	s22	d64	s23	d67	s24	d70	s25	d73	s26	d76
	d41	d42	d44	d45	d47	d48	d50	d51	d53	d54	d56	d57	d59	d60	d62	d63	d65	d66	d68	d69	d71	d72	d74	d75	d77	d78
1201	s27	d77	s28	d80	s29	d84	s30	d86	s31	d89	s32	d92	x1	x2	s32'	d92'	s31'	d89'	s30'	d86'	s29'	d83'	s28'	d80'	s27'	d77'
	d78	d79	d81	d82	d84	d85	d87	d88	d90	d91	p	p	x3	x4	p	p	d90'	d89'	d87'	d88'	d84'	d85'	d81'	d82'	d78'	d79'
1201	s26'	d76'	s25'	d73'	s24'	d70'	s23'	d67'	s22'	d64'	s21'	d61'	s20'	d58'	s19'	d55'	s18'	d52'	s17'	d49'	s16'	d46'	s15'	d43'	s14'	d40'
	d77'	d78'	d74'	d75'	d71'	d72'	d68'	d69'	d65'	d66'	d62'	d63'	d59'	d60'	d56'	d57'	d53'	d54'	d50'	d51'	d47'	d48'	d44'	d45'	d41'	d42'
1201	s13'	d37'	s12'	d34'	s11'	d31'	s10'	d28'	s9'	d25'	s8'	d22'	s7'	d19'	s6'	d16'	s5'	d13'	s4'	d10'	s3'	d7'	s2'	d4'	s1'	d1'
	d38'	d39'	d35'	d36'	d32'	d33'	d29'	d30'	d26'	d27'	d23'	d24'	d20'	d21'	d17'	d18'	d14'	d15'	d11'	d12'	d8'	d9'	d5'	d6'	d2'	d3'

FIG. 12A

00011000000000000011000...00011010000111111011011110010001001011100111110001010001

FIG. 12B

111001111111111100111...111001011110000001001000011011101101000110000001110101110

FIG. 12C

011010000111010001110100011110000001110100101...011101010100101001001100011100100101

FIG. 12D

10010111100010111000101110001111110001011010...100010101011010110110011100011011010

FIG. 12E

s1	d1	s2	d4	s3	d7	s4	d10	s5	d13	s6	d16	s7	d19	s8	d22	s9	d25	s10	d28	s11	d31	s12	d34	s13	d37
d2	d3	d5	d6	d8	d9	d11	d12	d14	d15	d17	d18	d20	d21	d23	d24	d26	d27	d29	d30	d32	d33	d35	d36	d38	d39
s14	d40	s15	d43	s16	d46	s17	d49	s18	d52	s19	d55	s20	d58	s21	d61	s22	d64	s23	d67	s24	d70	s25	d73	s26	d76
d41	d42	d44	d45	d47	d48	d50	d51	d53	d54	d56	d57	d59	d60	d62	d63	d65	d66	d68	d69	d71	d72	d74	d75	d77	d78
s27	d77	s28	d80	s29	d83	s30	d86	s31	d89	s32	d92	x	d95	s32'	d98	s31'	d101	s30'	d104	s29'	d107	s28'	d110	s27'	d113
d78	d79	d81	d82	d84	d85	d87	d88	d90	d91	d93	d94	d96	d97	d99	d100	d102	d103	d105	d106	d108	d109	d111	d112	d114	d115
s26'	d116	s25'	d119	s24'	d122	s23'	d125	s22'	d128	s21'	d131	s20'	d134	s19'	d137	s18'	d140	s17'	d143	s16'	d146	s15'	d149	s14'	d152
d117	d118	d120	d121	d123	d124	d126	d127	d129	d130	d132	d133	d135	d136	d138	d139	d141	d142	d144	d145	d147	d148	d150	d151	d153	d154
s13'	d155	s12'	d158	s11'	d161	s10'	d164	s9'	d167	s8'	d170	s7'	d173	s6'	d176	s5'	d179	s4'	d182	s3'	d185	s2'	d188	s1'	d191
d156	d157	d159	d160	d162	d163	d165	d166	d168	d169	d171	d172	d174	d175	d177	d178	d180	d181	d183	d184	d186	d187	d189	d190	d192	p

1310

FIG. 13

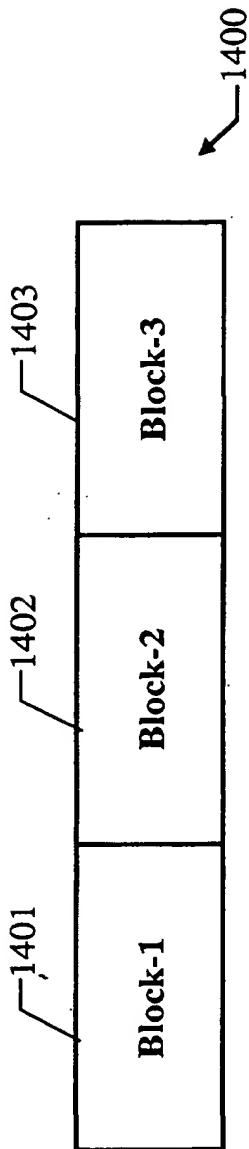


FIG. 14A

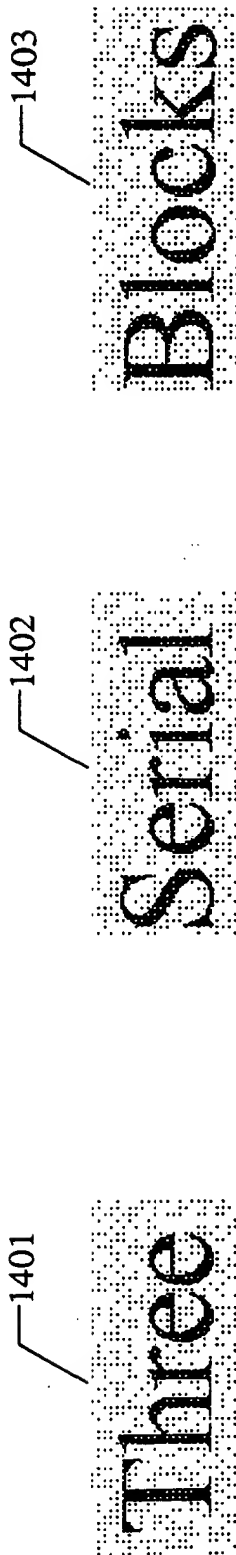


FIG. 14B

FIG. 14C

FIG. 14D

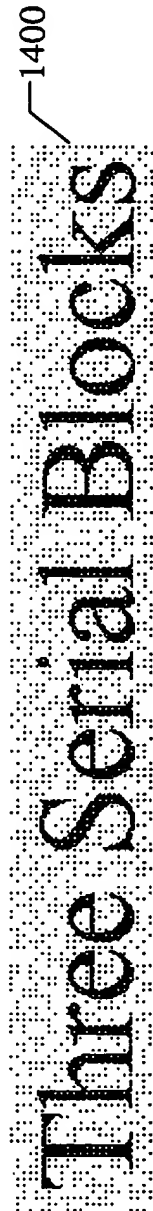


FIG. 14E

15/22

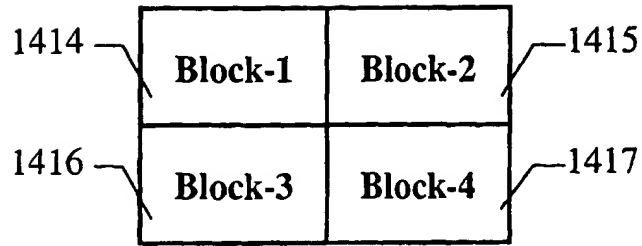


FIG. 14F

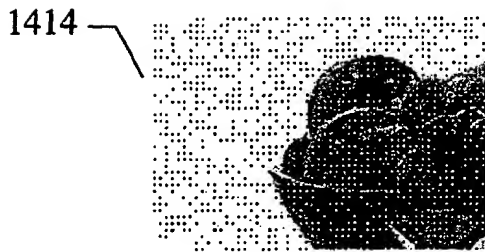


FIG. 14G

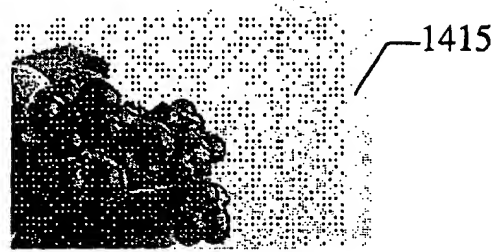


FIG. 14H

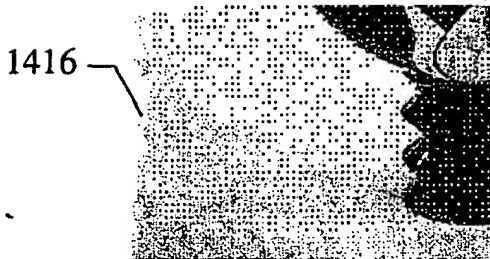


FIG. 14I

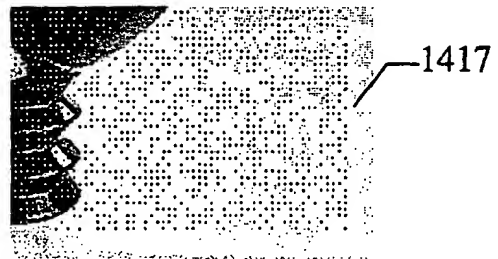


FIG. 14J

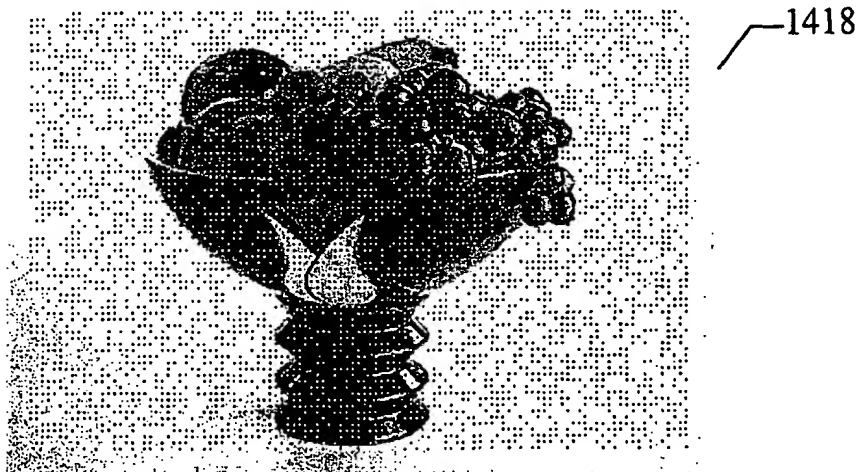


FIG. 14K

```
<HTML>
<BODY>
<P><A HREF="http://www.netshopper.com/"><B><FONT
FACE="Garamond" SIZE=6>Net Shopper</B></FONT></A></P>
<P><A HREF="http://www.netshopper.com/computers.html"><FONT
SIZE=5>Computers</FONT></A> </P>
<P><A HREF="http://www.netshopper.com/monitors.html"><FONT
SIZE=5>Monitors</FONT></A> </P>
<P><A HREF="http://www.netshopper.com/printers.html"><FONT
SIZE=5>Printers</FONT></A></P></BODY>
</HTML>
```

FIG. 15

1500

Net Shopper
Computers
Monitors
Printers

FIG. 16

1502

Net Shopper
Computers
Monitors
Printers

1506

1508

1510

1512

1504

FIG. 17

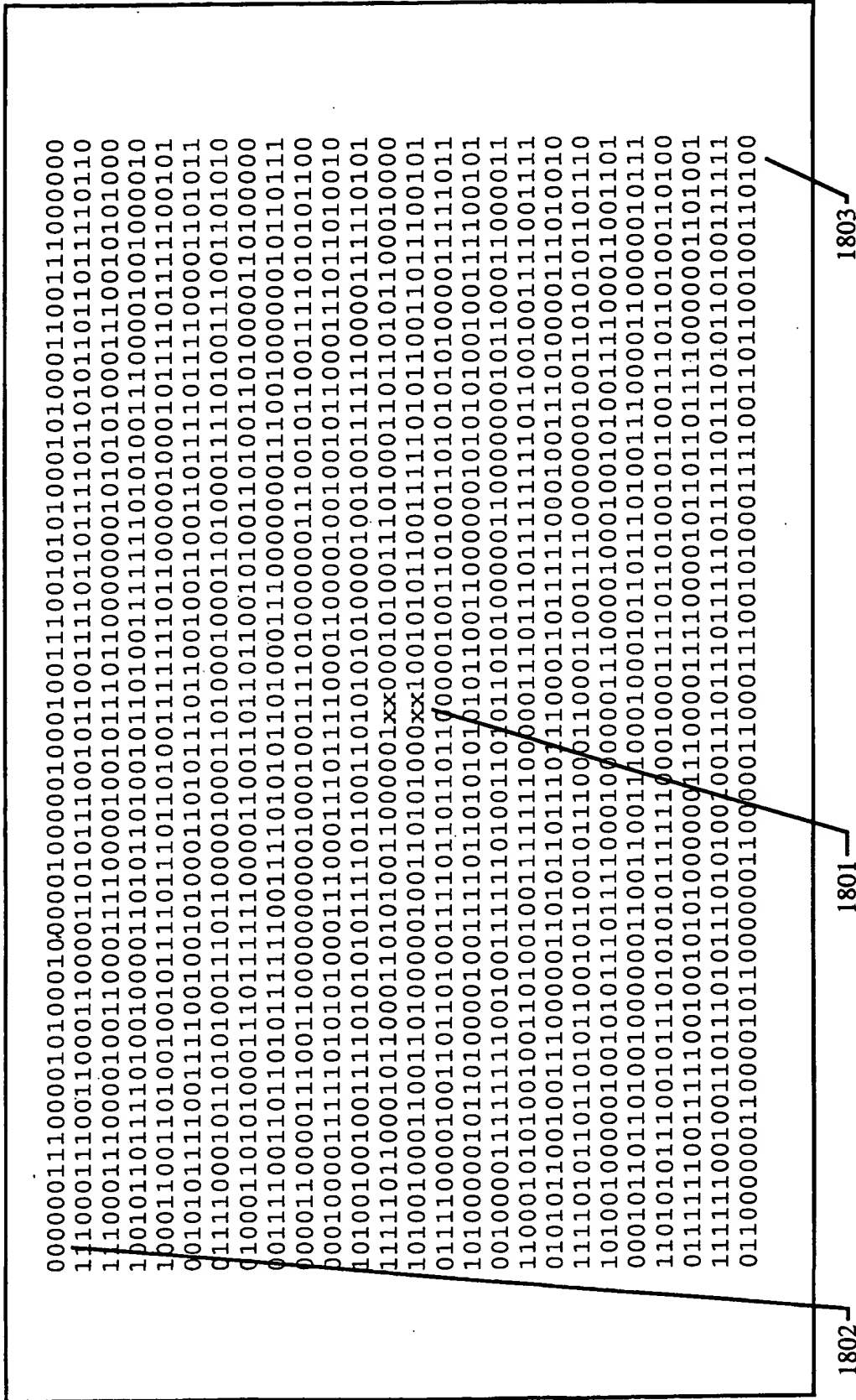


FIG. 18

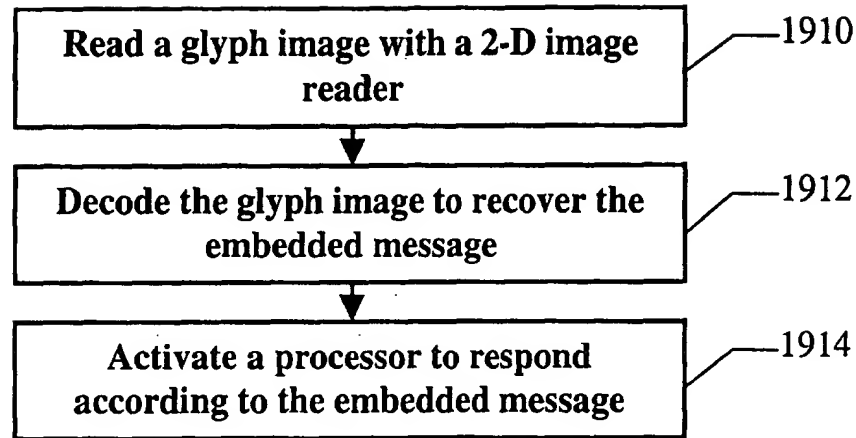


FIG. 19A

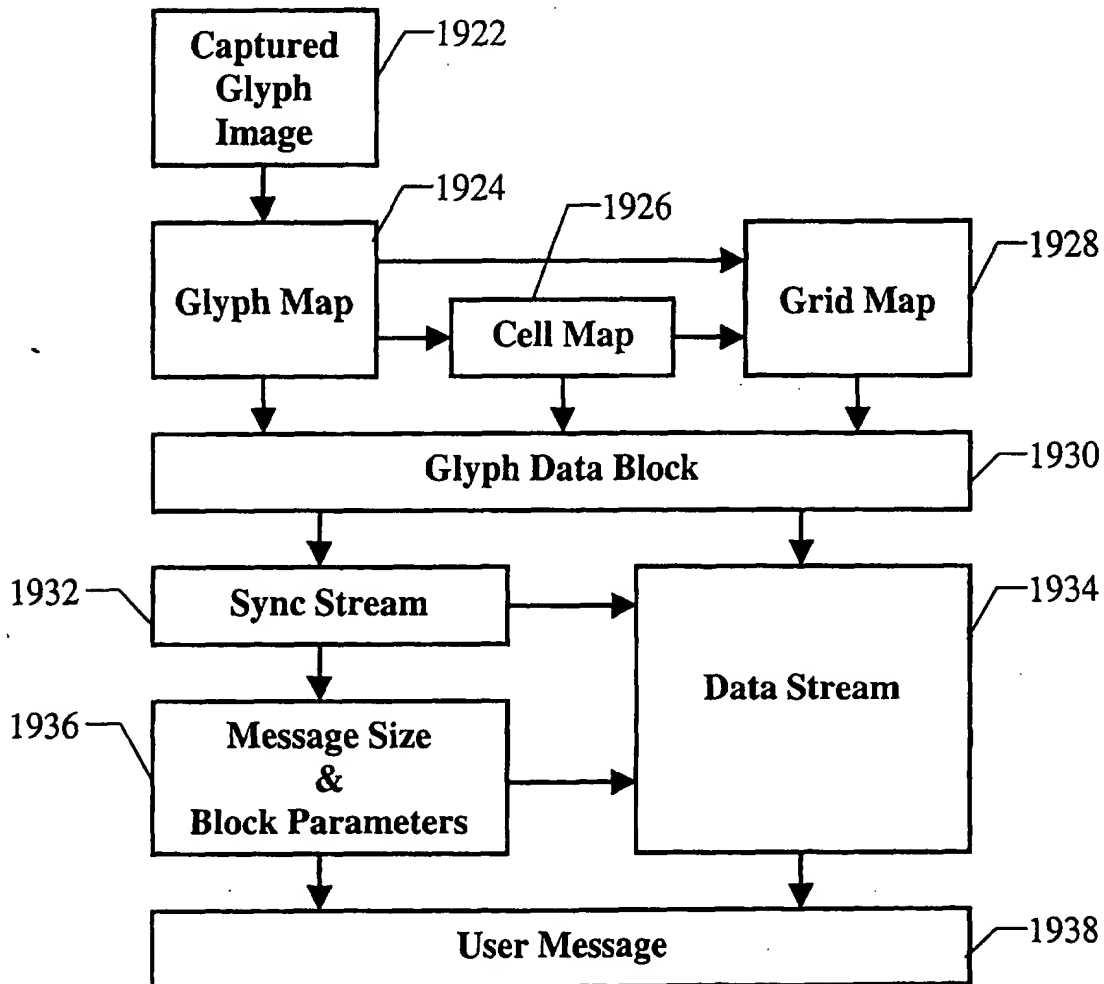


FIG. 19B

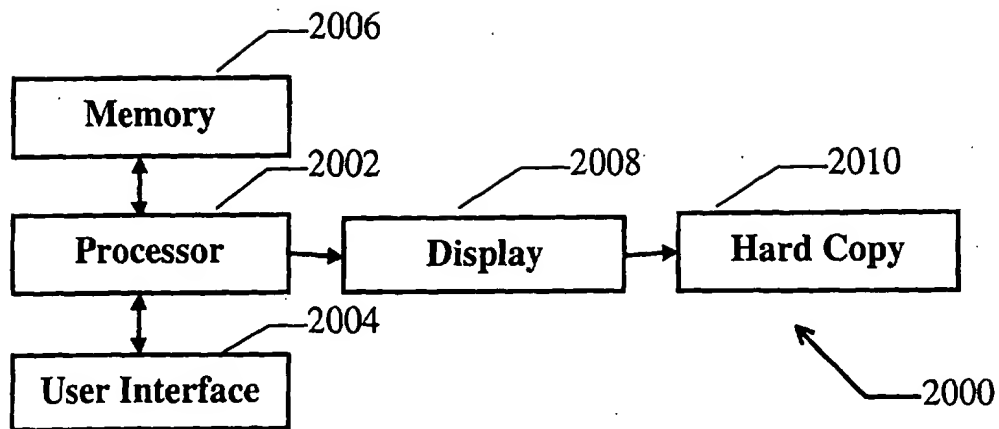


FIG. 20A

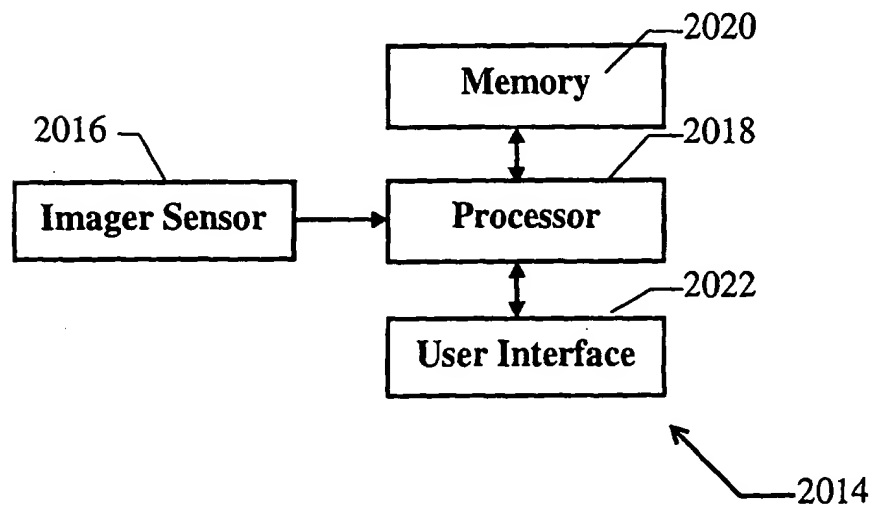


FIG. 20B

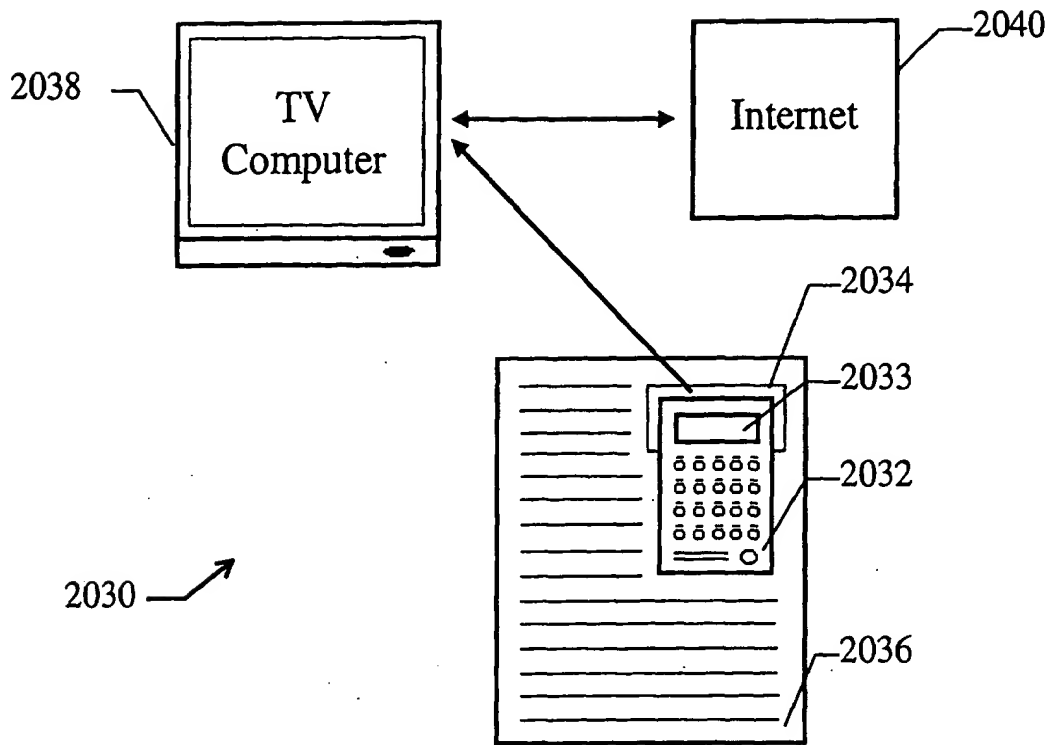


FIG. 20C

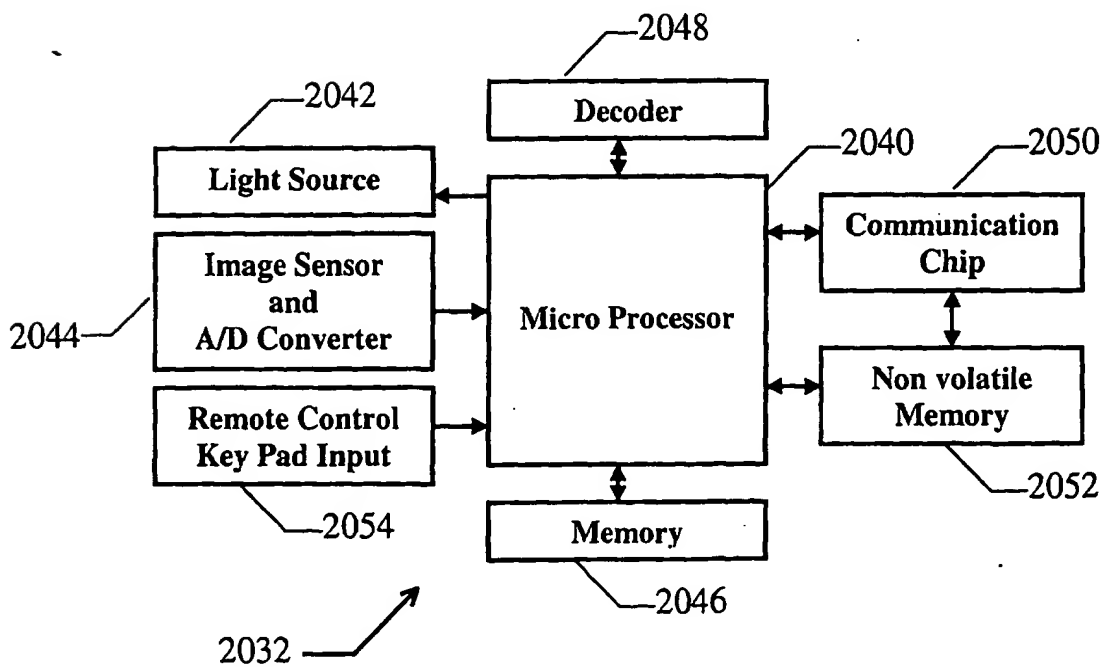


FIG. 20D

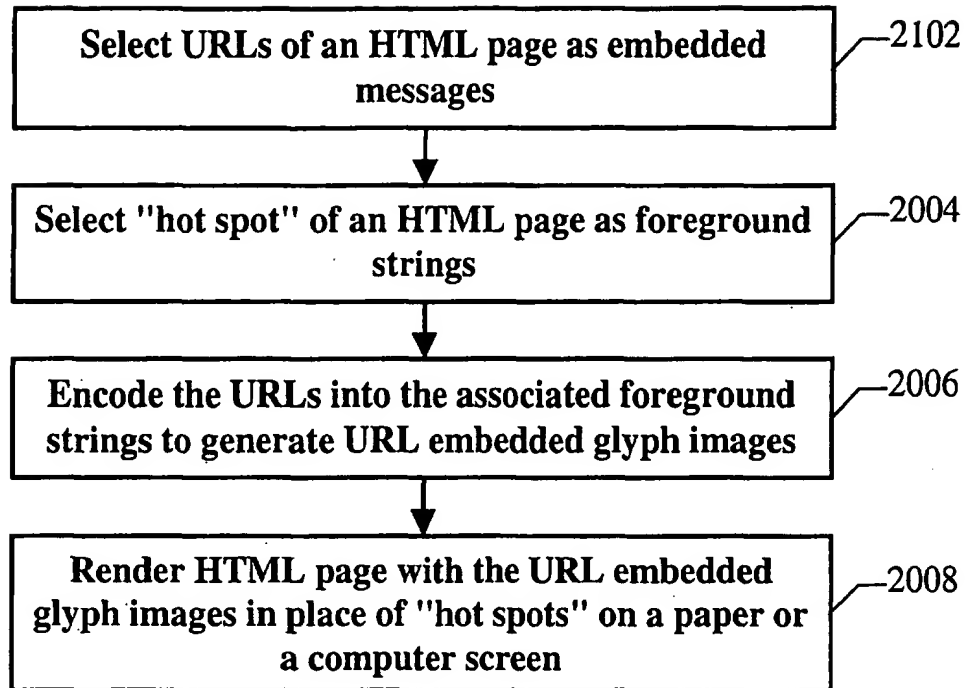


FIG. 21A

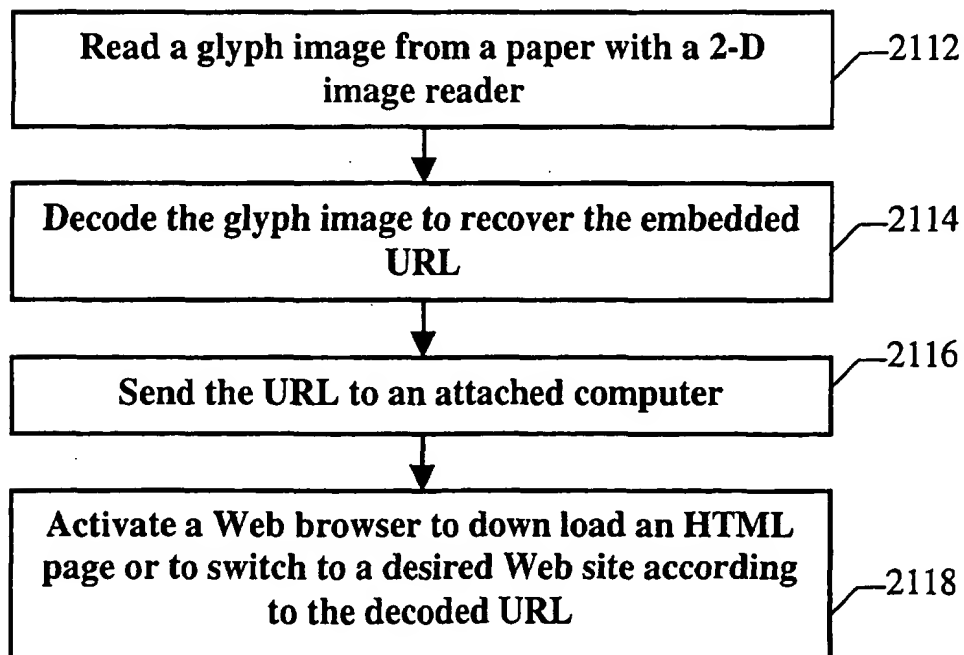


FIG. 21B

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☒ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☒ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.